



Universidad Pontificia Comillas

ICAI - Escuela Técnica Superior de Ingeniería

Trabajo Fin de Grado

---

***S'AforApp*: Aplicación para el control de  
aforo de las playas en Ibiza**

---

Grado en Ingeniería en Tecnologías de Telecomunicación

Autora: Adriana Tur Marí

Tutor: Atilano Ramiro Fernández-Pacheco Sánchez-Migallón

Madrid - Junio 2025



Declaro, bajo mi responsabilidad, que el Proyecto presentado con el título

**Aplicación para el control de aforo de las playas en Ibiza**

en la ETS de Ingeniería - ICAI de la Universidad Pontificia Comillas en el

curso académico 2024/25 es de mi autoría, original e inédito y

no ha sido presentado con anterioridad a otros efectos.

El Proyecto no es plagio de otro, ni total ni parcialmente y la información que ha sido

tomada de otros documentos está debidamente referenciada.



Fdo.: Adriana Tur Marí

Fecha: 30 / 06 / 2025

Autorizada la entrega del proyecto

EL DIRECTOR DEL PROYECTO

71216314B

ATILANO RAMIRO

FERNÁNDEZ-

PACHECO

Digitally signed by  
71216314B ATILANO  
RAMIRO FERNÁNDEZ-  
PACHECO  
Date: 2025.06.30 13:01:00  
+02'00'

Fdo.: Atilano Fernández-Pacheco Sánchez-Migallón Fecha: ...../ ...../ .....





Universidad Pontificia Comillas

ICAI - Escuela Técnica Superior de Ingeniería

Trabajo Fin de Grado

---

***S'AforApp: Aplicación para el control de  
aforo de las playas en Ibiza***

---

Grado en Ingeniería en Tecnologías de Telecomunicación

Autora: Adriana Tur Marí

Tutor: Atilano Ramiro Fernández-Pacheco Sánchez-Migallón

Madrid - Junio 2025



*A mi hogar, Ibiza, por haber sido la inspiración que dio vida a este proyecto.*



---

## Agradecimientos

Quiero agradecer a la Universidad y a todos los profesores de la carrera, por su humanidad y paciencia, por siempre estar dispuestos a repetir la explicación varias veces hasta que se entienda el concepto, por el cariño y la ilusión que le ponen a la docencia y por así hacer el camino mucho más ameno.

A mi padre, por transmitirme su pasión por las telecomunicaciones, enseñándome a contar en binario con tan solo 8 años. A mi madre, por enseñarme a gestionar emocionalmente ICAI, que todos sabemos que no es nada fácil. Y a ambos, por hacer posible con su apoyo (especialmente económico) que por fin pueda decir con orgullo que soy ingeniera y, por confiar en mí en todo momento, incluso cuando ni yo confiaba en mí misma.

A mi hermano Juan Carlos, por darme a conocer ICAI, por siempre ser un referente, una fuente de inspiración y sin duda el mejor ejemplo a seguir.

A Luis, por acompañarme incondicionalmente, estando presente los días buenos y los días no tan buenos y, estar siempre dispuesto a ayudar.

Y, sobre todo, gracias a mi hermana Silvia, por darme esa motivación y fuerza en los momentos más difíciles, porque sé que allí donde esté, está celebrando este logro conmigo.



---

# *S'AFORAPP*: APLICACIÓN PARA EL CONTROL DE AFORO DE LAS PLAYAS EN IBIZA

Autor: Tur Marí, Adriana.

Director: Fernández-Pacheco Sánchez-Migallón, Atilano Ramiro.

Entidad Colaboradora: ICAI - Universidad Pontificia Comillas.

## RESUMEN DEL PROYECTO

Este trabajo de fin de grado presenta el diseño y desarrollo de una aplicación para el control y gestión del aforo de las playas en Ibiza. La herramienta tiene como funcionalidades la consulta en tiempo real del nivel de ocupación de las playas, información que se obtiene gracias a la interacción de los usuarios. Además, incluye un sistema recomendador que propone alternativas en caso de que la playa buscada esté llena y un modelo de *Machine Learning* que predice la hora estimada de llenado en función de variables como la climatología o el día de la semana. El proyecto ha cumplido tres objetivos principales: implementar una aplicación funcional y accesible para la consulta del aforo, desarrollar un asistente inteligente que apoye la toma de decisiones, y contribuir a una distribución más equilibrada del turismo. Los resultados obtenidos favorecen una mejor organización de los visitantes, mejorando su experiencia en la isla y fomentando un turismo sostenible.

**Palabras clave:** Turismo Sostenible, Aplicación, *Machine Learning*, Aforo.

## 1. Introducción

Hoy en día, el turismo está en pleno auge y sobre todo en destinos como las islas Baleares y, concretamente, Ibiza. Por ello, la gestión eficiente de los espacios públicos como las zonas costeras es esencial para garantizar tanto una buena experiencia como la preservación del entorno. Con esta motivación, nace *s'AforApp*, este proyecto que ofrece una solución tecnológica y facilita la gestión del turismo de forma accesible y sostenible. La iniciativa parte del problema concreto de destinos de veraneo como la isla de Ibiza: la saturación de las playas en temporada alta que provoca saturaciones, impactos ambientales y una distribución desigual del turismo. De esta forma, el objetivo principal ha sido crear y aportar una herramienta que favorezca en la mitigación de estos efectos mediante el uso de las tecnologías y la participación del usuario para su correcto funcionamiento.

Antes de su desarrollo se han estudiado diversas iniciativas existentes. *Platges Segures*, Govern Balear (2020), impulsada durante la pandemia por el *Govern Balear*, ofrecía información manual sobre el aforo, pero sin funciones predictivas ni participación del usuario. *iPlaya*, proporciona datos meteorológicos y sobre servicios costeros, pero no contempla el control de ocupación. *PlayasApp*, Ayuntamiento de Málaga (2023), incluye cámaras en directo y geolocalización, aunque depende de dispositivos físicos y no integra predicción. Además, el *Consell Insular d'Eivissa* impulsa un sistema de cámaras y sensores ambientales para monitorizar playas (Vodafone España (2022)), aunque con retrasos en su implementación móvil y sin el factor colaborativo que caracteriza a *s'AforApp*.

Estas soluciones, aunque valiosas, presentan limitaciones comunes: falta de modelos

---

predictivos avanzados, escasa interacción con el usuario final y dependencia de infraestructuras. *S'AforApp* busca cubrir ese vacío, combinando tecnología, participación ciudadana y sostenibilidad en una propuesta adaptada a las necesidades de Ibiza.

## 2. Definición del proyecto

El proyecto, tal y como se ha mencionado anteriormente, consiste en la creación de una aplicación que permita visualizar el nivel de aforo de las playas en tiempo real, recibir recomendaciones en el caso de que la playa buscada se encuentre en su máximo aforo y prever el posible llenado de las playas en función de factores como el clima o el día de la semana. Todo esto, junto con otras funcionalidades secundarias, forman esta aplicación y para ello, el trabajo se ha dividido en tres grandes bloques:

1. El desarrollo de las funcionalidades principales de consulta y participación junto con las funcionalidades secundarias relacionadas con la gestión de las playas y la gestión de distintos perfiles de los usuarios.
2. La implementación de un sistema recomendador (“*Where to go*”).
3. El diseño de un modelo de predicción de la hora de llenado de las playas mediante *Machine Learning*.

## 3. Descripción del sistema

La arquitectura del sistema (figura 0.1) se compone de una aplicación (*frontend*), un servidor que gestiona las peticiones y almacena los datos (*backend*) y un modelo de aprendizaje automático entrenado con datos climatológicos simulados y APIs externas. Además de lo mencionado, se ha implementado una base de datos H2, que recoge en distintas tablas (figura 0.2) la información del aforo proporcionado por cada usuario, las playas de la isla, tanto los usuarios registrados como sus tokens de validación y la información de contacto y *feedback* recibido en la aplicación.

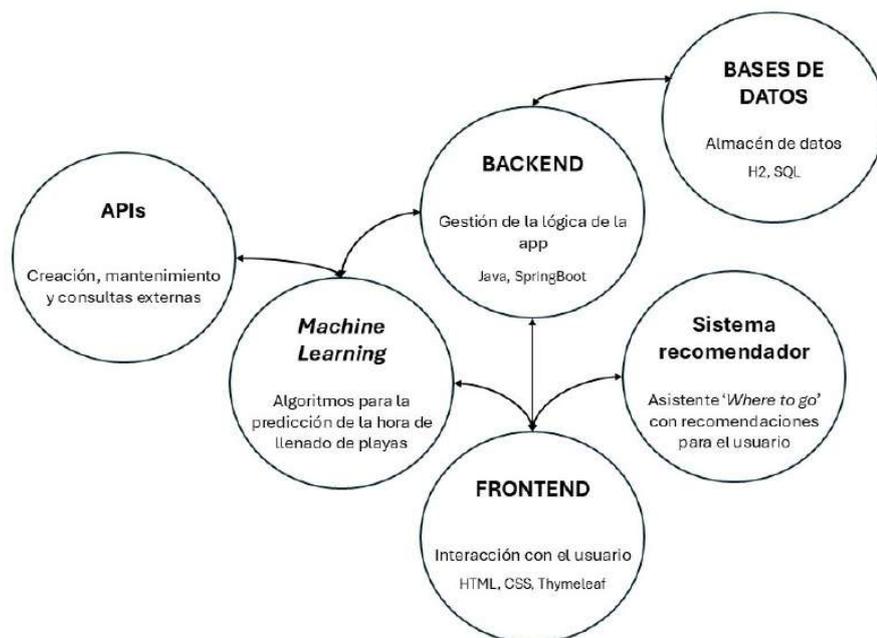


Figura 0.1: Arquitectura del sistema. Fuente: Elaboración propia

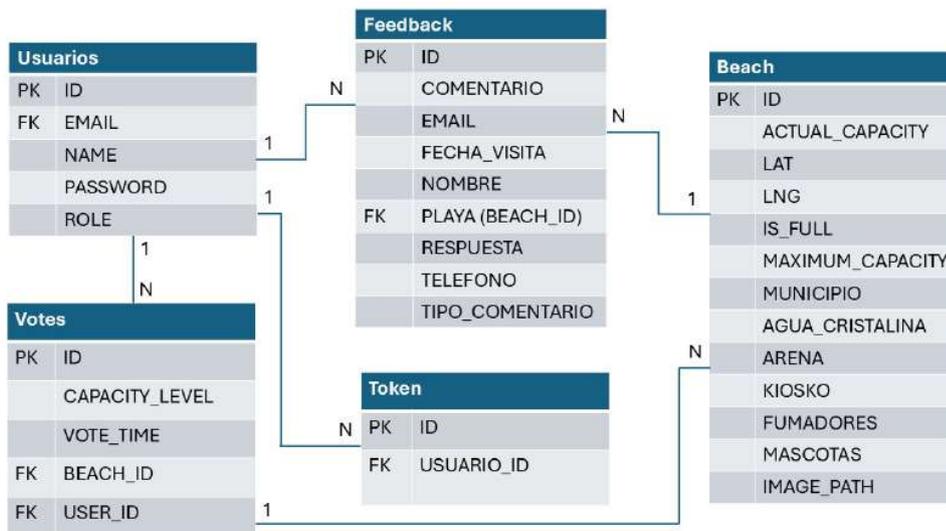


Figura 0.2: Diagrama entidad - relación. Fuente: Elaboración propia

#### 4. Resultados

Los resultados de este proyecto se han evaluado en función del cumplimiento de los objetivos definidos en la fase de planificación, ya que al tratarse del desarrollo de una aplicación el resultado sería la aplicación en sí. A continuación, se resumen los avances más relevantes vinculados con cada uno de los objetivos nombrados.

- **Objetivo 1. Gestión del aforo de las playas.**

Se ha creado una aplicación plenamente operativa que permite a los usuarios consultar el nivel de ocupación de las playas a través de una interfaz clara y adaptable a distintos dispositivos. El sistema distingue entre distintos tipos de usuario mediante autenticación, facilitando así tanto la consulta como la gestión de los distintos datos.

- **Objetivo 2. Algoritmo de predicción.**

Se ha integrado un modelo de predicción, realizado con *Machine Learning*, que estima cuándo una playa alcanzará su máximo aforo, utilizando variables climatológicas como la temperatura, la nubosidad o el viento para anticipar la hora de llenado.

- **Objetivo 3. Implementación del asistente “Where to go”.**

La aplicación incorpora un asistente que ofrece sugerencias alternativas cuando la playa de interés está completa, basándose en similitudes de características (tipo de arena, servicios...) y proximidad.

- **Objetivo 4. Fomento de un turismo sostenible.**

La combinación de información en tiempo real, sugerencias personalizadas y predicciones de ocupación permiten a los usuarios distribuirse de manera más eficiente entre las distintas playas de la isla, evitando aglomeraciones y traslados innecesarios. Esto busca mejorar la experiencia turística, contribuyendo a una gestión más equilibrada y sostenible de los espacios públicos y, en concreto, de las playas.

En la siguiente figura 0.3 se puede apreciar el menú de inicio de un Administrador:

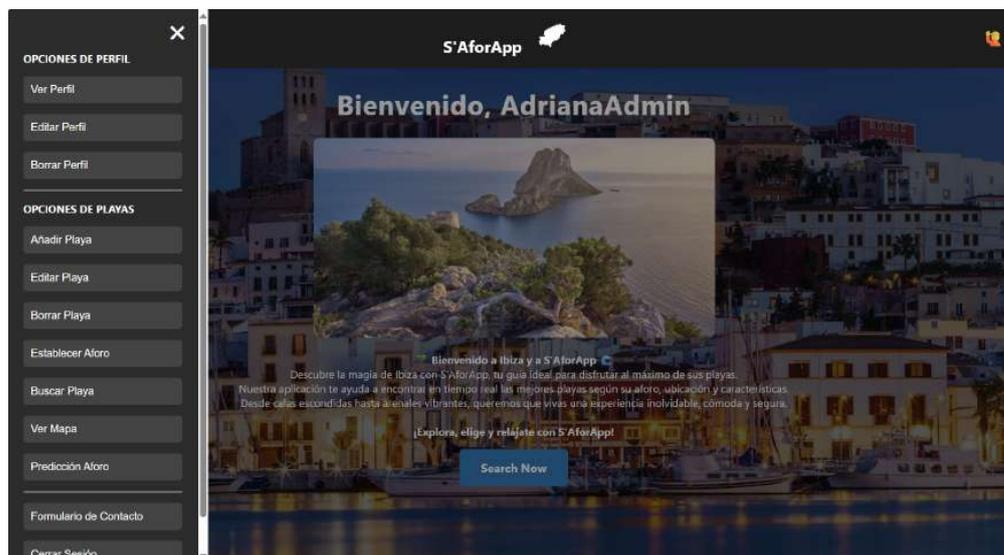


Figura 0.3: Menú de inicio de un Administrador. Fuente: Elaboración propia

## 5. Conclusiones

Se ha demostrado la viabilidad de aplicar tecnologías móviles e Inteligencia Artificial para mejorar la gestión del turismo en espacios naturales altamente concurridos y demandados. La aplicación desarrollada cubre satisfactoriamente los objetivos propuestos y sienta las bases para futuras ampliaciones. Entre las posibles líneas de trabajo se encuentran la incorporación de sensores externos para aumentar la precisión, la expansión a otras zonas geográficas, la adaptación a otros espacios públicos como parques naturales o la colaboración con entidades turísticas oficiales para su posible implantación real.

## 6. Referencias

- Govern Balear (2020). Platges segures: Aplicació de control d'aforament de platges.
- Vodafone España (2022). Ibiza instala cámaras con ia para controlar el aforo de 33 playas.
- Ayuntamiento de Málaga (2023). Playasapp: Consulta el estado de las playas.

---

# *S'AFORAPP*: APPLICATION FOR BEACH OCCUPANCY CONTROL IN IBIZA

Author: Tur Marí, Adriana.

Supervisor: Fernández-Pacheco Sánchez-Migallón, Atilano Ramiro.

Collaborating Entity: ICAI - Comillas Pontifical University.

## ABSTRACT

This final degree project presents the design and development of an application for monitoring and managing beach occupancy in Ibiza. The tool includes functionalities such as real-time consultation of beach crowd levels, with information provided through user interaction. Additionally, it incorporates a recommendation system that suggests alternative beaches when the selected one is full, and a Machine Learning model that predicts the estimated time of saturation based on variables such as weather conditions or the day of the week. The project has achieved three main objectives: implementing a functional and accessible application for occupancy consultation, developing an intelligent assistant to support decision-making, and contributing to a more balanced distribution of tourism. The results obtained support better organization of visitors, enhancing their experience on the island and promoting sustainable tourism.

**Keywords:** Sustainable tourism, Application, Machine Learning, Occupancy.

## 1. Introduction

Nowadays, tourism is experiencing significant growth, especially in destinations like the Balearic Islands, and in particular, Ibiza. For this reason, the efficient management of public spaces, such as coastal areas, is essential to ensure both a positive visitor experience and the preservation of the environment. With this motivation, *s'AFORAPP* was born—a project that offers a technological solution to support accessible and sustainable tourism management.

The initiative arises from a specific issue faced by holiday destinations like Ibiza: overcrowded beaches during the high season, which leads to congestion, environmental degradation, and an uneven distribution of tourism. The main goal, therefore, has been to create and provide a tool that helps mitigate these effects through the use of technology and user participation to ensure its proper operation.

Before its development, several existing initiatives were studied. *Platges Segures*, Govern Balear (2020), launched during the pandemic by the Balearic Government, offered manually updated beach occupancy information but lacked predictive capabilities and user engagement. *iPlaya* provides weather forecasts and coastal service information, but does not monitor beach occupancy. *PlayasApp*, Ayuntamiento de Málaga (2023), includes live camera access and geolocation features, though it depends on physical monitoring devices and lacks predictive functionality. Additionally, the *Consell Insular d'Eivissa* is promoting a system of cameras and environmental sensors to monitor beaches (Vodafone España (2022)), but its mobile application has been delayed and lacks the collaborative element that defines *s'AFORAPP*.

While these solutions are valuable, they share common limitations: a lack of advanced predictive models, minimal user interaction, and heavy reliance on physical infrastruc-

---

ture. *S'Aforapp* aims to address these gaps by combining technology, citizen participation, and sustainability in a solution tailored to the specific needs of Ibiza.

## 2. Project definition

As previously mentioned, the project consists of the creation of an application that allows users to visualize the real-time occupancy levels of beaches, receive recommendations when the selected beach is at maximum capacity, and anticipate potential crowding based on factors such as weather conditions or the day of the week. These features, along with additional secondary functionalities, make up the core of the application. To structure the development, the project has been divided into three main components:

1. Development of the core features for occupancy consultation and user participation, along with secondary functionalities related to beach management and user profile handling.
2. Implementation of a recommendation system (“Where to go”).
3. Design of a predictive model to estimate beach saturation times using Machine Learning.

## 3. System description

The system architecture (figure 0.4) is composed of an application (frontend), a server that handles requests and manages data storage (backend), and a machine learning model trained with simulated and weather-related data from external APIs. In addition to these components, an H2 database has been implemented to store information on beach occupancy provided by users, as well as data related to beaches, users, tokens and the feedback received in the app, in structured tables (figure 0.5).

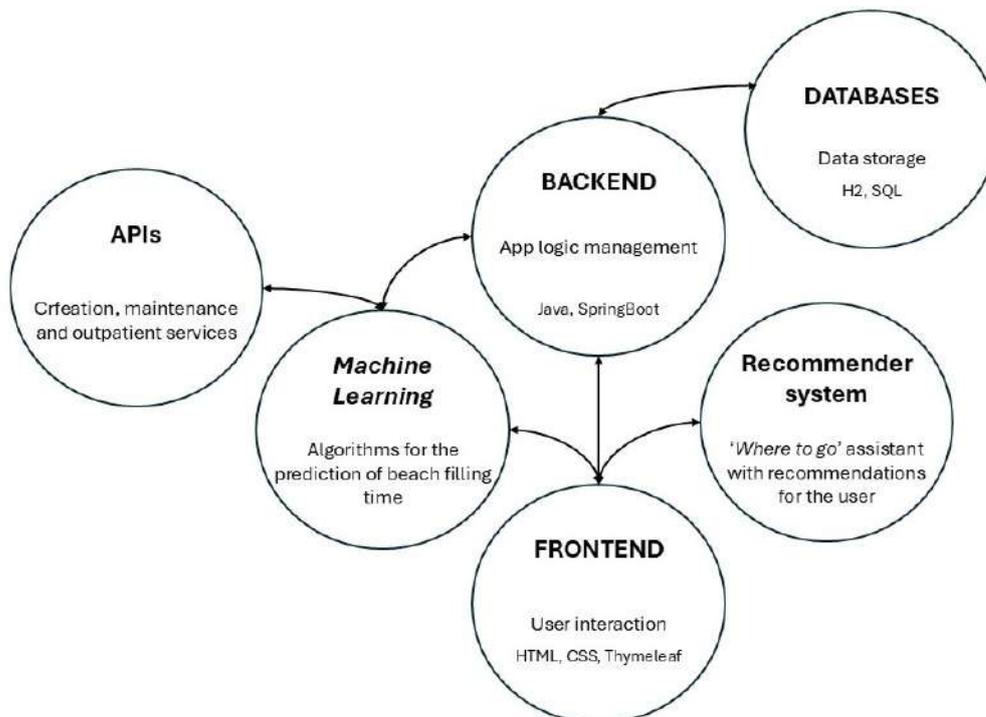


Figura 0.4: System architecture. Source: Own elaboration

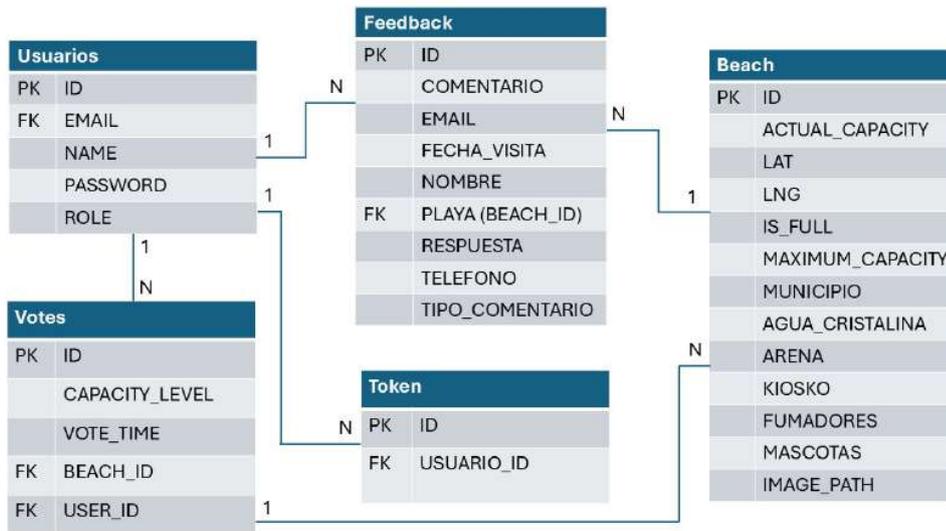


Figura 0.5: Entity - relationship diagram. Source: Own elaboration

#### 4. Results

The results of this project have been evaluated based on the achievement of the objectives defined during the planning phase, as the outcome—being the development of an application—is the application itself. Below is a summary of the most relevant progress made in relation to each stated objective:

- **Objective 1. Beach capacity management.**

A fully functional application has been created, allowing users to check the current occupancy levels of beaches through a clear and responsive interface. The system supports different user roles via authentication, enabling both data consultation and administrative management.

- **Objective 2. Prediction algorithm.**

A predictive model based on *Machine Learning* has been integrated to estimate when a beach will reach its maximum occupancy, using weather-related variables such as temperature, cloudy or windy to anticipate crowding times.

- **Objective 3. Implementation of the “Where to go” assistant.**

The application includes an assistant that suggests alternative beaches when the selected one is at full capacity, based on similarity in features (sand, services...) and proximity.

- **Objective 4. Promotion of sustainable tourism.**

By combining real-time information, personalized recommendations, and predictive capabilities, the application helps users distribute themselves more evenly across the island’s beaches. This prevents overcrowding and reduces unnecessary travel, aiming to enhance the tourist experience while contributing to a more balanced and sustainable management of public spaces—particularly coastal areas.

The following image 0.6 shows the home menu of an Administrator:



Figura 0.6: Administrator Home Menu. Source: Own elaboration

## 5. Conclusions

The project has demonstrated the feasibility of applying mobile technologies and artificial intelligence to improve tourism management in natural environments. The developed application successfully meets the proposed objectives and establishes a solid foundation for future enhancements. Potential lines of future work include the integration of external sensors to increase accuracy, expansion to other geographic areas, its use in other public areas like natural parks and collaboration with official tourism entities for possible real-world implementation.

## 6. References

- Govern Balear (2020). Platges segures: Aplicació de control d'aforament de platges.
- Vodafone España (2022). Ibiza instala cámaras con ia para controlar el aforo de 33 playas.
- Ayuntamiento de Málaga (2023). Playasapp: Consulta el estado de las playas.

---

# ÍNDICE GENERAL

<b>Índice de figuras</b>	<b>XXIII</b>
<b>Índice de cuadros</b>	<b>XXV</b>
<b>Índice de siglas</b>	<b>XXVII</b>
<b>1. Introducción</b>	<b>1</b>
1.1. Motivación del proyecto . . . . .	1
1.2. Resultado del proyecto . . . . .	4
1.3. Estructura de la memoria . . . . .	5
<b>2. Tecnologías y Herramientas Empleadas</b>	<b>7</b>
2.1. <i>Spring Boot</i> . . . . .	7
2.2. JPA ( <i>Java Persistence API</i> ) . . . . .	8
2.3. Base de datos H2 . . . . .	9
2.4. JSON . . . . .	9
2.5. REST . . . . .	10
2.6. <i>Jakarta Bean Validation</i> . . . . .	10
2.7. <i>Spring Security</i> . . . . .	11
2.8. <i>Maven</i> . . . . .	11
2.9. Modelos de <i>Machine Learning</i> . . . . .	12
2.10. Consumo de APIs . . . . .	12
2.10.1. Propias . . . . .	12
2.10.2. Externas . . . . .	13
2.11. Mapas e información geográfica . . . . .	13
2.12. Servidor de correo automático . . . . .	14
2.13. <i>Github</i> . . . . .	14
<b>3. Estado de la Cuestión</b>	<b>15</b>
3.1. Aplicaciones existentes en el mercado . . . . .	15
3.2. Proyectos de investigación y literatura científica existentes . . . . .	18

<b>4. Definición del Trabajo</b>	<b>19</b>
4.1. Objetivos	19
4.1.1. Gestión del aforo de las playas	19
4.1.2. Desarrollo de un algoritmo que prediga la hora de llenado de las playas	20
4.1.3. Desarrollo de un asistente inteligente ‘Where to go’	20
4.1.4. Fomento de un turismo sostenible	21
4.2. Metodología	21
4.3. Planificación y estimación económica	22
<b>5. Justificación y Desarrollo del Sistema</b>	<b>25</b>
5.1. Análisis del sistema	26
5.1.1. Requisitos funcionales	26
5.1.2. Requisitos no funcionales	30
5.2. Diseño, Implementación y Justificación	32
5.2.1. Interfaz de usuario ( <i>fronted</i> )	33
5.2.2. Servidor ( <i>backend</i> )	34
5.2.2.1. Funciones principales	35
5.2.2.2. Tecnologías utilizadas	36
5.2.3. Base de datos	36
5.2.4. Servicios meteorológicos externos	40
5.2.5. Sistemas predictivos ( <i>Machine Learning</i> )	40
5.2.6. Seguridad	43
5.2.7. Diagrama de componentes	44
5.2.8. Justificación práctica y social	44
5.2.9. Diferencias frente al mercado actual	45
5.2.9.1. Turismo sostenible	46
<b>6. Análisis de Resultados</b>	<b>47</b>
6.1. Objetivo 1: Gestión del aforo de las playas	47
6.2. Objetivo 2: Algoritmo para la predicción de la hora de llenado de las playas	48
6.3. Objetivo 3: Desarrollo del asistente inteligente ‘Where to go’	48
6.4. Objetivo 4: Fomento de un turismo sostenible	48
<b>7. Conclusiones y Trabajos Futuros</b>	<b>51</b>
7.1. Conclusiones	51
7.2. Trabajos futuros	52
<b>A. Alineación del Proyecto con los ODS</b>	<b>I</b>
<b>B. Manual de usuario de <i>S’AforApp</i></b>	<b>V</b>
B.1. Introducción	V
B.2. Acceso a la Aplicación	VI
B.2.1. Registro de Usuario	VII
B.2.2. Inicio de sesión	VII
B.3. Funcionalidades principales	VIII
B.3.1. Consultar Aforo de Playas	IX

## ÍNDICE GENERAL

---

B.3.1.1.	Búsqueda de playas y filtrado de playas . . . . .	IX
B.3.1.2.	Recomendador de playas ( <i>'Where to go'</i> ) . . . . .	X
B.3.1.3.	Visualización de mapa . . . . .	XI
B.3.2.	Establecer Aforo . . . . .	XIII
B.3.3.	Predicción de hora de llenado . . . . .	XIII
B.3.4.	Gestión del perfil . . . . .	XV
B.3.4.1.	Visualizar perfil . . . . .	XV
B.3.4.2.	Editar perfil . . . . .	XV
B.3.4.3.	Borrar perfil . . . . .	XVI
B.3.5.	Formulario de contacto . . . . .	XVII
B.3.6.	Cerrar sesión . . . . .	XVIII
B.4.	Funciones avanzadas (únicamente para administradores) . . . . .	XVIII
B.4.1.	Gestión de playas . . . . .	XVIII
B.4.1.1.	Añadir playa . . . . .	XVIII
B.4.1.2.	Modificar playa . . . . .	XIX
B.4.1.3.	Eliminar playa . . . . .	XX
B.4.2.	Editar perfil (tipo de usuario) . . . . .	XX
B.5.	Recomendaciones de uso . . . . .	XX
B.6.	Soporte . . . . .	XXI
<b>Bibliografía</b>		<b>XXIII</b>



---

# ÍNDICE DE FIGURAS

0.1. Arquitectura del sistema. Fuente: Elaboración propia . . . . .	XII
0.2. Diagrama entidad - relación. Fuente: Elaboración propia . . . . .	XIII
0.3. Menú de inicio de un Administrador. Fuente: Elaboración propia . . . .	XIV
0.4. System architecture. Source: Own elaboration . . . . .	XVI
0.5. Entity - relationship diagram. Source: Own elaboration . . . . .	XVII
0.6. Administrator Home Menu. Source: Own elaboration . . . . .	XVIII
1.1. Situación geográfica de las Islas Baleares. Fuente: <i>Wikipedia</i> . . . . .	2
1.2. Número de turistas (en millones) en las Islas Baleares entre 2008 y 2023. Fuente: <i>Statista</i> . . . . .	2
1.3. Logo de la aplicación resultante, <i>S'AforApp</i> . Fuente: Elaboración propia	5
2.1. Logo de <i>Spring Boot</i> . Fuente: Página oficial de <i>Spring Boot</i> . . . . .	8
3.1. Logo de <i>iPlaya</i> . Fuente: Página oficial de <i>iPlaya</i> . . . . .	16
4.1. Planificación temporal del trabajo. Fuente: Elaboración propia . . . . .	22
5.1. Arquitectura del sistema. Fuente: Elaboración propia . . . . .	26
5.2. Tabla de Usuarios, base de datos. Fuente: Elaboración propia . . . . .	37
5.3. Tabla de Playas, base de datos. Fuente: Elaboración propia . . . . .	37
5.4. Tabla de Votaciones, base de datos. Fuente: Elaboración propia . . . . .	38
5.5. Tabla de <i>Feedback</i> , base de datos. Fuente: Elaboración propia . . . . .	38
5.6. Tabla de <i>Token</i> , base de datos. Fuente: Elaboración propia . . . . .	39
5.7. Diagrama entidad-relación, base de datos. Fuente: Elaboración propia .	39
5.8. Diagrama de componentes. Fuente: Elaboración propia . . . . .	44
A.1. Resumen de los 17 ODS. Fuente: Página web de la ONU . . . . .	II
B.1. Pantalla de inicio dinámica. Fuente: Elaboración propia . . . . .	VI
B.2. Pantalla con opciones de 'Inicio de Sesión' y 'Registro'. Fuente: Elabora- ción propia . . . . .	VI
B.3. Formulario de registro. Fuente: Elaboración propia . . . . .	VII
B.4. Inicio de sesión. Fuente: Elaboración propia . . . . .	VII

B.5. Menús de inicio. Fuente: Elaboración propia . . . . .	VIII
B.6. Filtro del buscador de playas. Fuente: Elaboración propia . . . . .	IX
B.7. Buscador de playas. Fuente: Elaboración propia . . . . .	X
B.8. Recomendador de playas. Fuente: Elaboración propia . . . . .	XI
B.9. Visualización del mapa de las playas de Ibiza. Fuente: Elaboración propia	XII
B.10. Funcionalidad de establecer aforo. Fuente: Elaboración propia . . . . .	XIII
B.11. Predicción de la hora de llenado de las playas mediante un modelo de <i>Machine Learning</i> . Fuente: Elaboración propia . . . . .	XIV
B.12. Página de visualización del perfil. Fuente: Elaboración propia . . . . .	XV
B.13. Página de edición del perfil. Fuente: Elaboración propia . . . . .	XVI
B.14. Página de borrado del perfil. Fuente: Elaboración propia . . . . .	XVI
B.15. Formulario de contacto para recibir <i>feedback</i> del usuario. Fuente: Elabo- ración propia . . . . .	XVII
B.16. Página de adición de playas. Fuente: Elaboración propia . . . . .	XIX
B.17. Página de edición de playas. Fuente: Elaboración propia . . . . .	XIX
B.18. Página de borrado de playas. Fuente: Elaboración propia . . . . .	XX

---

# ÍNDICE DE CUADROS

4.1. Estimación de costes del proyecto. Fuente: Elaboración propia . . . . .	23
5.1. Métricas de los mejores algoritmos. Fuente: Elaboración propia . . . . .	42
5.2. Métricas de los mejores modelos de predicción. Fuente: Elaboración propia	43



---

# ÍNDICE DE SIGLAS

AEMET	Agencia Estatal de Meteorología
AJAX	<i>Asynchronous JavaScript And XML</i>
API	<i>Application Programming Interface</i>
CORS	<i>Cross-Origin Resource Sharing</i>
CSS	<i>Cascading Style Sheets</i>
DL	<i>Deep Learning</i>
GPS	<i>Global Positioning System</i>
HTML	<i>HyperText Markup Language</i>
HTTP	<i>Hypertext Transfer Protocol</i>
ID	<i>Identification</i>
IDE	<i>Integrated Development Environment</i>
INE	Instituto Nacional de Estadística
IA	Inteligencia Artificial
IUV	Índice Ultravioleta
JDBC	<i>Java Database Connectivity</i>
JDK	<i>Java Development Kit</i>
JSON	<i>JavaScript Object Notation</i>
JPI	<i>JAVA Persistence API</i>
JVM	<i>Java Virtual Machine</i>
ML	<i>Machine Learning</i>
MAE	<i>Mean Absolute Error</i>
MSE	<i>Mean Squared Error</i>
RNR	Redes Neuronales Recurrentes
REST	<i>Representational State Transfer</i>
SMTP	<i>Simple Mail Transfer Protocol</i>
SQL	<i>Structured Query Language</i>

## Índice de siglas

---

ONU	.....	Organización de las Naciones Unidas
UTC	.....	<i>Universal Time Coordinated</i>
URL	.....	<i>Uniform Resource Locator</i>
UTE	.....	Unión Temporal de Empresas
UX	.....	<i>User Experience</i>

---

---

# CAPÍTULO 1

---

## INTRODUCCIÓN

En el presente capítulo se ofrece una introducción y contextualización del proyecto con el fin de situar al lector. Se presenta una breve introducción del proyecto realizado, exponiendo la motivación que lo impulsa (sección 1.1), el resultado alcanzado (sección 1.2) y la estructura que organiza los distintos puntos de esta memoria (sección 1.3).

### 1.1. Motivación del proyecto

Las Islas Baleares son un archipiélago español situado en el Mar Mediterráneo de gran fama tanto nacional como internacional. Es la elección de muchos turistas como su destino de veraneo y cuenta con cuatro islas principales: Mallorca, Menorca, Ibiza y Formentera (figura 1.1).

Tal es la afluencia de visitantes que, según informa el diario *EuropaPress*<sup>1</sup> (basándose en datos del INE, Instituto Nacional de Estadística), las Islas Baleares cerraron el año 2024 con más de 18,7 millones de turistas. Esta cifra la sitúa por encima de grandes ciudades turísticas como Barcelona, Dubái, Rio de Janeiro, Berlín o Shanghai y muy cerca de otras urbes punteras en turismo como Tokio, con 19 millones de visitantes en 2024.

---

<sup>1</sup>Baleares cierra el 2024 por encima de los 18,7 millones de turistas

Este valor supone un incremento del 5% con respecto al año anterior, consolidando una tendencia alcista (figura 1.2) que se ha recuperado notablemente después de la reducción de turistas en los años vividos durante la pandemia de la Covid-19 en todo el mundo.

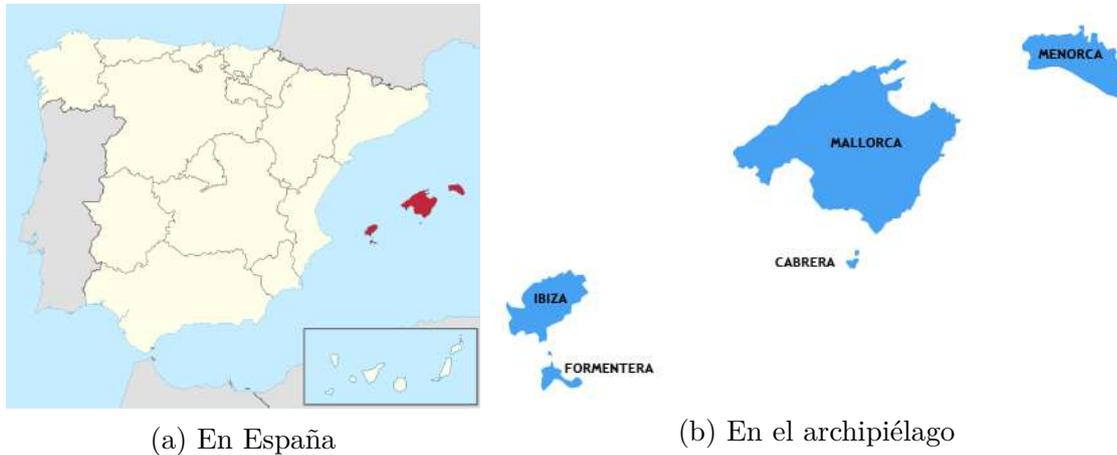


Figura 1.1: Situación geográfica de las Islas Baleares. Fuente: *Wikipedia*

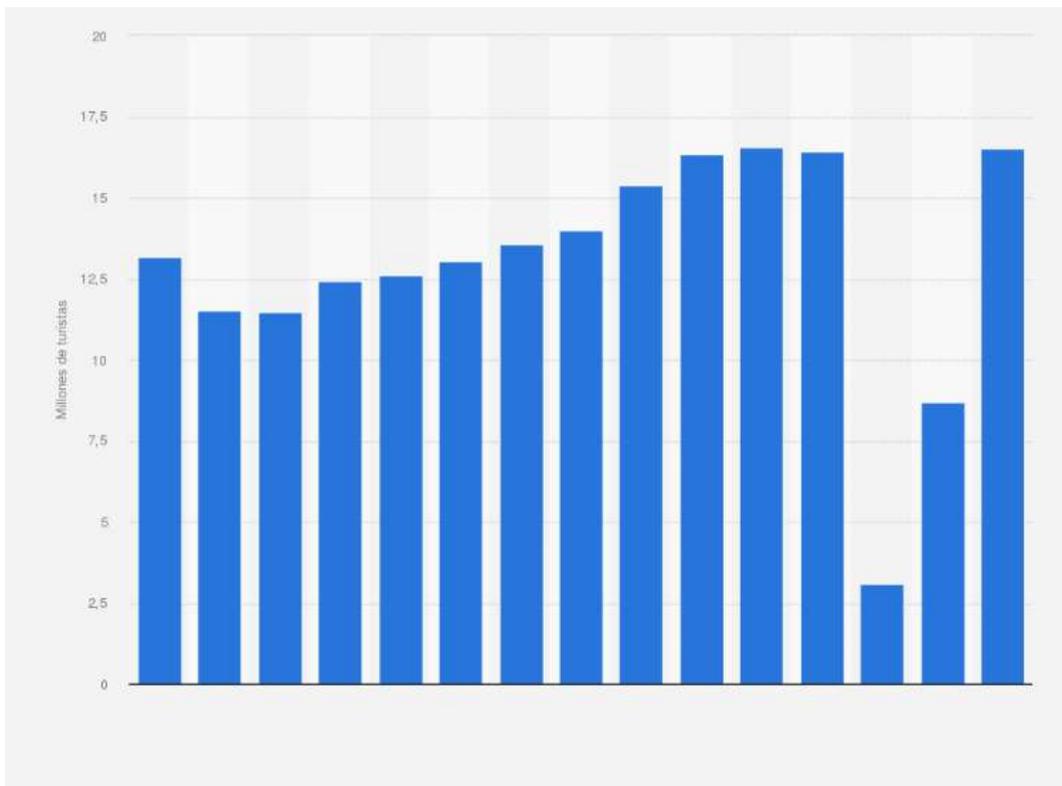


Figura 1.2: Número de turistas (en millones) en las Islas Baleares entre 2008 y 2023. Fuente: *Statista*

La isla de Ibiza es una de las principales del archipiélago balear y se ha convertido

en un destino turístico de renombre, conocido tanto dentro de España como fuera de nuestras fronteras. Año tras año, millones de visitantes acuden a la isla para disfrutar de la belleza de sus paisajes, playas y atardeceres, su clima, su patrimonio cultural e historia y, también hay que mencionarlo, su reconocida vida nocturna que le ha dado fama mundial a la isla.

Pero esta popularidad tiene un coste. Con una población que no llega a los 150.000 habitantes, Ibiza llega a multiplicar por doce el número de personas en sus picos de mayor afluencia turística, coincidiendo con el periodo estival y de vacaciones. Esta presión tan elevada provoca una serie de problemas que no solo afectan a la calidad de vida de quienes viven allí todo el año, sino también a la experiencia de quienes la visitan.

El crecimiento del turismo en Ibiza ha traído consigo varios desafíos y uno de los más importantes es la saturación que sufre la isla, sobre todo durante los meses de verano (Granados (2024)). La concentración de un número tan elevado de personas en un territorio limitado afecta directamente a las infraestructuras y a los recursos naturales, que se ven sometidos a una presión constante y difícil de sostener a largo plazo. Esta situación no solo pone en peligro el equilibrio medioambiental, sino que también ha provocado un fuerte encarecimiento del mercado inmobiliario y del coste de vida, haciendo cada vez más complicado para los residentes encontrar una vivienda asequible.

Como consecuencia de esta gran afluencia de visitantes, se vive una masificación permanente de los espacios públicos y de las playas, un fenómeno que no solo genera molestias entre la población local, sino que también afecta negativamente a la experiencia de los propios turistas. Ante este panorama, las autoridades han comenzado a tomar medidas para intentar frenar la presión turística. Una de las más recientes ha sido la restricción de entrada de vehículos a la isla, en vigor desde el 1 de junio de 2025<sup>2</sup>, siguiendo el modelo aplicado anteriormente en Formentera, con la intención de reducir el impacto del turismo masivo y avanzar hacia una gestión más sostenible.

Además del comentado impacto social y económico, el turismo descontrolado también deja huella en el medioambiente. La acumulación de residuos y deshechos en las playas

---

<sup>2</sup>Ibiza intenta aliviar la saturación turística limitando la entrada de coches

(que llega a representar hasta el 80% de la basura marina en verano según un estudio del *Institut de Ciència i Tecnologia Ambientals (ICTA-UAB) (2021)*), y el aumento del tráfico en busca de zonas menos concurridas están afectando seriamente al entorno natural. Por eso, se ha convertido en una prioridad el control del número de visitantes y la limitación del uso de vehículos con el objetivo de avanzar hacia un turismo más sostenible y respetuoso con el medioambiente.

Esta realidad hace que surga la necesidad de crear una solución tecnológica que ayude a gestionar mejor dicho flujo de turistas. Con ello, no solo se mejorará su experiencia en la isla, sino que también se cuidará el medioambiente, garantizando así la sostenibilidad de la isla en el tiempo.

En este sentido, se propone la aplicación ***S'AforApp***, una herramienta pensada para que tanto gestores, turistas y residentes puedan decidir fácilmente qué playa visitar. Esta solución, que se explica en el siguiente apartado, busca distribuir mejor a las personas y promover un turismo más equilibrado y respetuoso con el entorno.

### 1.2. Resultado del proyecto

Tal y como se introdujo en el apartado anterior con la motivación del trabajo propuesto (sección 1.1), el proyecto se centra en el desarrollo de una aplicación denominada ***S'AforApp***, cuyo objetivo principal es gestionar de manera eficiente y sostenible el flujo turístico en las playas de la isla de Ibiza. Esta aplicación promueve la interacción directa con el usuario, permitiendo obtener datos en tiempo real sobre el aforo de cada playa gracias a la participación ciudadana. El logotipo de la *app* se muestra en la figura 1.3.

La información sobre el aforo de las distintas playas de la isla se almacena en una base de datos, que es actualizada constantemente gracias a la información proporcionada por los propios usuarios, permitiendo así mostrar datos precisos y actualizados de forma inmediata.

Entre sus principales funcionalidades destacan la posibilidad de consultar y votar el

nivel de ocupación de las playas, así como acceder a un mapa interactivo con imágenes e información contextual. Además, ya se ha comentado la existencia un sistema de predicción del aforo para anticipar niveles de ocupación y planificar visitas de forma más eficiente.

Todas las playas recogidas en la base de datos se muestran ordenadas en función de su cercanía a la ubicación actual del usuario, facilitando así el acceso a opciones próximas. Asimismo, para cada playa se ofrece información detallada sobre sus características, como el tipo de arena, la claridad del agua, la presencia de kioscos, la posibilidad de fumar o acudir con mascotas, entre otros aspectos relevantes para el visitante.



Figura 1.3: Logo de la aplicación resultante, *S'AforApp*. Fuente: Elaboración propia

### 1.3. Estructura de la memoria

La memoria del proyecto se divide en 7 capítulos que detallarán su realización:

- En el capítulo 1 se ha realizado un breve resumen del proyecto, incluyendo la motivación de este, una introducción y esta breve descripción de como se realizará la memoria.
- En el capítulo 2 se describirán las tecnologías, protocolos, herramientas específicas,

etc. con el fin de facilitar la lectura de la memoria, que incluye de forma recurrente términos técnicos aquí explicados.

- El capítulo 3 detalla el estado del arte, recogiendo soluciones y herramientas que ya existen en el ámbito del proyecto en otras localizaciones.
- En el capítulo 4 se detallarán los objetivos así como una explicación de la metodología y planificación y estimación económica del trabajo llevado a cabo.
- El capítulo 5 se justificará la necesidad de la implementación del proyecto, a la vez que se profundiza en el propio proyecto, realizando un análisis del mismo y abordando su diseño e implementación.
- En el capítulo 6 se detallarán los resultados obtenidos.
- En el capítulo 7 indicará las conclusiones y sugerirá futuros trabajos a realizar para mejorar y escalar la aplicación.

Además, esta memoria cuenta con dos anexos que completan el trabajo:

- El anexo A profundiza en la relación de este proyecto con los Objetivos de Desarrollo Sostenible (ODS) y cómo puede la aplicación fomentar un mundo más justo y mejor para todos.
- El anexo B es un manual para el usuario donde se presentan todas las funcionalidades de la aplicación de forma gráfica y sencilla.

---

---

# CAPÍTULO 2

---

## TECNOLOGÍAS Y HERRAMIENTAS EMPLEADAS

En este capítulo se describen las tecnologías y herramientas que han sido fundamentales para el desarrollo del proyecto. La intención es que cualquier lector, incluso sin formación técnica, pueda entender de forma sencilla qué tecnologías se han usado, para qué sirven y cómo se relacionan entre sí. A lo largo de esta memoria aparecerán repetidamente nombres como *Spring Boot*, JPA o REST, por lo que es importante explicar desde el principio qué significan y por qué se han elegido.

Cada una de estas tecnologías cumple un papel específico dentro de la aplicación, y juntas hacen posible que el sistema funcione de forma fluida, segura y eficiente.

### 2.1. *Spring Boot*

La primera de ellas es *Spring Boot*, un conjunto de herramientas que facilita enormemente el desarrollo de aplicaciones en Java, uno de los lenguajes de programación más utilizados en el mundo. Se podría decir que *Spring Boot* es similar a una caja de herramientas inteligente que ayuda al usuario a construir una aplicación web sin tener que preocuparse por configurar todo desde cero.

Antes de su existencia, el desarrollo de aplicaciones basadas en Java requería una configuración manual extensa, incluyendo la creación y ajuste de múltiples archivos, así como la escritura de una cantidad considerable de código inicial para lograr que la aplicación estuviera operativa. Con *Spring Boot*, gran parte de esa preparación inicial ya viene resuelta de forma predefinida y automatizada, lo que permite a los desarrolladores centrarse directamente en crear las funcionalidades específicas de su aplicación.

Este marco de trabajo también promueve una estructura ordenada del código, haciendo que sea más fácil mantener y entender el sistema a largo plazo. Además, se integra de forma muy sencilla con otras herramientas que también se han utilizado en este proyecto, como las bases de datos o los servicios web, garantizando la seguridad como uno de los pilares fundamentales de la aplicación.



Figura 2.1: Logo de *Spring Boot*. Fuente: Página oficial de *Spring Boot*

### 2.2. JPA (*Java Persistence API*)

JPA (por sus siglas en inglés *Java Persistence API*) es una especificación estándar para trabajar con bases de datos desde el lenguaje de programación Java, asegurando la persistencia de los datos. Permite que el programador guarde, modifique o consulte información en la base de datos sin tener que escribir sentencias en SQL (por sus siglas en inglés *Structured Query Language*), lenguaje que normalmente se usa para interactuar con las bases de datos.

En lugar de trabajar con código SQL explícito, el desarrollador interactúa con objetos Java y JPA se encarga de traducir esas acciones en instrucciones que las bases de datos puedan entender, simplificando así el desarrollo y facilitando la portabilidad del código creado.

## 2.3. Base de datos H2

H2<sup>1</sup> es una base de datos muy ligera y rápida que funciona directamente en la memoria del ordenador. Esto significa que es ideal para las fases de desarrollo y pruebas, ya que no requiere instalación ni configuración adicional. Cada vez que se ejecuta el sistema, se crea una base de datos temporal que se borra al apagar la aplicación.

Este enfoque es muy útil cuando se está programando, ya que permite probar cambios de forma inmediata y sin riesgos. En un entorno de producción (es decir, cuando la aplicación ya está en uso real), H2 puede ser sustituida fácilmente por bases de datos más potentes como *MySQL* o *PostgreSQL* sin necesidad de realizar modificaciones o adaptaciones al código.

## 2.4. JSON

JSON<sup>2</sup> (por sus siglas en inglés *JavaScript Object Notation*) es el formato en el que se intercambian los datos dentro de la aplicación, específicamente entre el *frontend* y el *backend*. Su estructura es muy similar al lenguaje humano, lo que lo hace fácil de entender tanto para los desarrolladores como para los sistemas informáticos. Además, permite almacenar todo tipos de datos: texto, numéricos (enteros o decimales), *booleans*...

En esta aplicación, por ejemplo, cada playa se representa mediante un objeto JSON que contiene todas sus características relevantes, como el nombre, el aforo, la ubicación o el tipo de agua, entre otros atributos:

```
{  
  "playa": "Aigües Blanques",  
  "longitud": 39.0595,  
  "latitud": 1.58951,  
  "aforo": "medio",  
  "kiosko": True,  
}
```

---

<sup>1</sup>Documentación oficial de H2

<sup>2</sup>Documentación oficial de JSON

## 2.5. REST

REST (por sus siglas en inglés *Representational State Transfer*) es un estilo de arquitectura que define cómo deben comunicarse los distintos componentes de una aplicación a través de internet. En este proyecto, se ha adoptado por el modelo REST para permitir la interacción entre la interfaz de usuario (*frontend* o lo que el usuario ve) y la lógica del servidor (*backend*), facilitando así el intercambio de datos y operaciones.

Esta arquitectura emplea *URLs* (direcciones web) para localizar recursos, junto con métodos estandarizados HTTP (por sus siglas en inglés *Hypertext Transfer Protocol*) como *GET* (recuperar datos), *POST* (enviar datos), *PUT* (actualizar recursos) y *DELETE* (borrar recursos). Esto permite que los distintos módulos de la aplicación se comuniquen de forma clara y estandarizada.

## 2.6. Jakarta Bean Validation

Cuando un usuario introduce datos en una aplicación (por ejemplo, al registrar una playa o al votar cuán llena está), es importante comprobar que esa información es válida antes de guardarla.

Para ello, se ha utilizado *Jakarta Bean Validation*<sup>3</sup>, una tecnología que permite aplicar reglas automáticas de validación a los objetos de la aplicación (guardados en JSON) mediante anotaciones como por ejemplo:

- `@NotNull`: asegura que un campo no esté vacío. Por ejemplo, a la hora de registrar un nuevo usuario en la aplicación, es imprescindible que éste posea un nombre para identificarlo a la hora de realizar futuras acciones como *logearse* o votar el aforo de una playa.
- `@Email`: comprueba que el valor introducido por el usuario sigue la estructura de un correo electrónico, con el uso del caracter `@`.
- `@Size(min=3, max=50)`: limita la cantidad de caracteres de un texto. Empleado en el registro de contraseñas para exigir una longitud mínima.

---

<sup>3</sup>Documentación oficial de *Jakarta Validation*

Estas validaciones ayudan a evitar errores, mantener la calidad de los datos y mejorar la experiencia del usuario.

### 2.7. *Spring Security*

Para evitar que usuarios no autorizados accedan a ciertas partes del sistema o realicen acciones para las cuales no poseen permisos, se ha implementado *Spring Security*<sup>4</sup>.

Este *framework* permite:

- **Autenticación:** comprobar la identidad de quien intenta acceder. Solo aquellos usuarios registrados correctamente podrán acceder al contenido de la aplicación.
- **Autorización:** verificar qué acciones puede realizar esa persona. La aplicación cuenta con dos perfiles de usuarios: usuarios simples o administradores y poseen algunas diferencias en cuanto a las acciones que pueden realizar (por ejemplo, la eliminación de una playa del sistema solo está permitido a los administradores de la aplicación, nunca a los usuarios).
- **Protección contra accesos no autorizados y vulnerabilidades.**

Esta capa de seguridad es vital cuando se manejan datos personales, identidades y funcionalidades críticas.

### 2.8. *Maven*

*Maven*<sup>5</sup> es una herramienta que ayuda a gestionar y automatizar proyectos de software desarrollados principalmente en Java. En concreto, se encarga de:

- Descargar automáticamente todas las bibliotecas necesarias.
- Compilar el código fuente y ejecutar pruebas.
- Generar versiones listas para instalar o subir al servidor.

Con Maven, un equipo de desarrolladores se asegura de poder trabajar con la misma configuración y evitar errores por versiones incompatibles. También facilita la organización del proyecto.

---

<sup>4</sup>Documentación oficial de *Spring Security*

<sup>5</sup>Documentación oficial de *Maven*

## 2.9. Modelos de *Machine Learning*

Una de las funciones más avanzadas del proyecto es la predicción del aforo en las playas con el fin de que el usuario pueda planificar sus desplazamientos a las playas en futuros días. Para ello se ha utilizado un algoritmo de *Machine Learning*, rama de la Inteligencia Artificial, que se ha entrenado con datos históricos simulados para facilitar su aprendizaje y poder tener predicciones futuras lo más reales posibles. Por ello, se han creado numerosos modelos, uno para cada una de las playas registradas.

Aunque en capítulos posteriores se explicará más en detalle (subsección 5.2.5), el modelo toma como datos de entrada, para cada playa, el día de la semana y mes sobre el que se realizará la predicción así como datos básicos sobre las condiciones meteorológicas previstas (temperatura máxima y mínima, porcentaje de nubosidad o intensidad y dirección del viento, entre otros). En base a todos estos datos, devuelve una hora de llenado estimada de la playa.

## 2.10. Consumo de APIs

La aplicación consume dos tipos de APIs (por sus siglas en inglés *Application Programming Interface*) de las que obtener información para su explotación. Estos dos tipos de APIs se pueden dividir en APIs propias y APIs externas o de terceros.

### 2.10.1. Propias

Una vez entrenados los distintos modelos de *Machine Learning* para la predicción del aforo en las playas, estos se han guardado en archivos con formato .pkl (*Pickle*, formato propio de *Python* para la serialización de objetos) para facilitar su consulta por parte del sistema.

Esta consulta se realiza mediante una API propia desarrollada también en *Python*, de forma que cuando el usuario solicita la predicción, el sistema accede a ella y consume los modelos almacenados (en función de la playa seleccionada por el usuario, usará el modelo correspondiente) para generar y devolver una respuesta al *frontend*.

### 2.10.2. Externas

Para la predicción de la hora de llenado por parte del modelo, la aplicación necesita acceder a información del clima previsto en tiempo real, como la temperatura, la previsión de lluvia o la hora de la puesta de sol. Para obtener estos datos, se conecta a APIs externas, que son servicios disponibles en internet que proporcionan información actualizada.

Estas conexiones se hacen desde el *backend* utilizando peticiones HTTP, y los datos se reciben en formato JSON para que puedan ser procesados y enviados al modelo predictivo.

Para el proyecto, se han empleado dos APIs externas:

- ***OpenWeatherMap API***<sup>6</sup>: servicio que proporciona datos meteorológicos en tiempo real y pronósticos climáticos para diferentes ubicaciones alrededor del mundo. Para las predicciones, se obtienen los datos en la isla de Ibiza para la fecha solicitada por el usuario.
- ***Sunrise and sunset API***<sup>7</sup>: ofrece datos precisos sobre los horarios de salida y puesta de sol, así como la duración del día y la noche, para cualquier ubicación geográfica especificada. Esto es fundamental para calcular el número de horas de sol para el modelo.

## 2.11. Mapas e información geográfica

Para ofrecer una experiencia más personalizada, la aplicación emplea el sistema de coordenadas GPS (por sus siglas en inglés *Global Positioning System*) para identificar y mostrar las playas más cercanas al usuario. La geolocalización es especialmente útil para turistas y residentes que buscan alternativas menos concurridas cerca de su ubicación actual.

Gracias a esta funcionalidad:

- El sistema puede ordenar las playas según la proximidad al usuario.

---

<sup>6</sup>API de *OpenWeatherMap*

<sup>7</sup>API de *Sunrise and sunset*

- El usuario puede ver ubicaciones exactas en un mapa.
- Se facilita la planificación del viaje a la playa más adecuada.

### 2.12. Servidor de correo automático

La aplicación cuenta con un apartado donde el usuario puede contactar mediante un formulario con la organización para realizar quejas, sugerencias o comentarios. En este apartado, se emplea un servidor de correo automático cuya funcionalidad es enviar, de forma automática, una copia del formulario relleno al correo que se haya indicado como garantía de su recepción por la parte administradora de la aplicación. Esta funcionalidad ha sido implementada desde el *backend* mediante la configuración de un servicio SMTP (por sus siglas en inglés *Simple Mail Transfer Protocol* o Protocolo Simple de Transferencia de Correo en castellano).

### 2.13. *GitHub*

Durante el desarrollo del proyecto se ha utilizado *GitHub*, una plataforma que permite almacenar y controlar los cambios en el código fuente mediante un sistema llamado control de versiones, permitiendo:

- Guardar el historial completo del proyecto.
- Recuperar versiones anteriores si ocurre un error.
- Trabajar en equipo de forma eficiente, integrando los cambios de varios desarrolladores (en el caso de que los hubiera).

Usar *GitHub* mejora la organización y la eficiencia del trabajo y proporciona una copia segura del proyecto.

---

---

# CAPÍTULO 3

---

## ESTADO DE LA CUESTIÓN

Antes de iniciar el desarrollo de una aplicación tecnológica orientada a la gestión del aforo en playas, resulta necesario realizar un estudio sobre las soluciones existentes, tanto a nivel comercial como académico.

El resultado de este análisis se presenta en este capítulo, identificando algunas de las aplicaciones existentes relevantes en el mercado, valorando sus enfoques y objetivos y analizando posibles carencias en otros proyectos que justifiquen así la creación y desarrollo del proyecto propuesto.

### 3.1. Aplicaciones existentes en el mercado

En los últimos años han surgido diversas iniciativas dirigidas a mejorar la gestión y difusión de información en las playas. Aunque éstas surgieron inicialmente como respuesta a la pandemia mundial de Covid-19, una vez disminuyó la preocupación relacionada con el uso de los espacios públicos, esta idea se puede desarrollar y aprovechar especialmente en destinos con un alto volumen turístico como puede ser la isla de Ibiza.

Uno de los primeros ejemplos en el ámbito nacional fue *Platges Segures*, una plataforma impulsada por el Govern de *les Illes Balears* en 2020, Ajuntament de Palma

(2020). Esta aplicación buscaba la posibilidad de consultar el aforo en las playas, con el fin de prevenir las grandes aglomeraciones para evitar así contagios en las playas. Los datos que se consultaban habían sido introducidos generalmente de forma manual por personal municipal, haciendo necesario la contratación de personal para su mantenimiento.

El uso de esta aplicación fue de corta duración ya que se propuso con un fin preventivo y para ayudar a frenar la pandemia y, no tenía ni funcionalidades predictivas ni de interacción con el usuario final, Govern Balear (2020).

Otro caso destacable es **iPlaya** (figura 3.1), una *app* que ofrece una guía de más de 2.000 playas entre España, Portugal e Italia. Actualmente está disponible para su consumo y proporciona información sobre servicios (duchas, eventos, aseos, alquileres etc.) y pronóstico del tiempo para los próximos 10 días, extraído de la AEMET (Agencia Estatal de Meteorología), como la temperatura del aire y del agua, las mareas, el oleaje, el IUUV (Índice Ultravioleta) o el viento, entre otros. Pero no permite la consulta en tiempo real del aforo de las playas.



Figura 3.1: Logo de *iPlaya*. Fuente: Página oficial de *iPlaya*

A nivel local, algunos municipios del litoral mediterráneo han desarrollado soluciones como **PlayasApp**, que proporciona al usuario información geográfica, acceso a cámaras en directo y estado de las playas. Aunque supone un buen recurso, esta aplicación no consta de un modelo de predicción de aforo y, a diferencia de la aplicación propuesta, su funcionalidad depende de dispositivos de monitorización instalados físicamente,

Ayuntamiento de Málaga (2023).

Esto complica su puesta en marcha por motivos logísticos y económicos así como una posible escalabilidad y transportabilidad a otros entornos y ubicaciones.

Por último, cabe destacar que el *Consell Insular d'Eivissa* está llevando a cabo un proyecto para el control de aforo en las playas de la mayor de las Pitiusas. Este proyecto forma parte de la iniciativa “Ibiza, Turismo Inteligente y Sostenible”, impulsada por el *Consell* en colaboración con Red.es (empresa pública dependiente del Ministerio de Asuntos Económicos y Transformación Digital) y llevada a cabo por la UTE (Unión Temporal de Empresas) Vodafone-Kapsch, Vodafone España (2022). El objetivo de este proyecto es monitorizar en tiempo real la ocupación de las playas mediante la instalación de cámaras y el uso de tecnologías avanzadas, Periódico de Ibiza (2022).

En cuanto a la instalación de dispositivos tecnológicos, se han colocado cámaras en un total de 33 playas de Ibiza con el fin de capturar instantáneas y así calcular el porcentaje de ocupación. Estos datos recopilados se recogen en una plataforma digital que permite visualizar la información sobre el aforo y la cual es accesible de forma pública. También, se planeaba desarrollar una aplicación móvil, la cual estaba prevista para antes del final del verano de 2022 pero varios problemas técnicos han supuesto un retraso en su lanzamiento.

Además del control de las playas, este proyecto también incluye sensores con el fin de medir la calidad del aire o de prevenir y detectar de forma rápida incendios y así mejorar la sostenibilidad y la seguridad en la isla Balear. Este último enfoque es bastante interesante y novedoso ya que dadas las altas temperaturas en verano, los incendios son una gran preocupación en la isla.

Estos ejemplos previamente explicados, muestran que la digitalización del turismo costero es un objetivo claro, pero también evidencian limitaciones y carencias comunes: la mayoría de las soluciones actuales ofrecen información unidireccional, carecen de sistemas predictivos basados en Inteligencia Artificial, y no contemplan la interacción activa del usuario como fuente de datos.

## 3.2. Proyectos de investigación y literatura científica existentes

En el ámbito académico, existen investigaciones relacionadas con la predicción de afluencia de personas en espacios públicos mediante modelos de Aprendizaje Automático. Por ejemplo, estudios como el de Luo et al. (2020) aplican modelos de *Deep Learning* como las Redes Neuronales Recurrentes (RNN) para estimar flujos de visitantes en parques urbanos, considerando datos históricos y meteorológicos. Aunque el caso de estudio de este artículo está basado en contextos urbanos, esta misma lógica podría ser aplicada a playas o espacios naturales.

También existen trabajos que abordan el uso de *crowdsourcing* (proceso de colaboración abierta y distribuida donde se delega una tarea a un grupo grande y diverso de personas) como mecanismo de recogida de datos en tiempo real, destacando el potencial de la colaboración ciudadana para generar información útil en entornos donde no se dispone de sensores físicos, Kanhere (2013). Esta línea de trabajo se alinea con la lógica de la aplicación propuesta, ya que busca la interacción con el usuario mediante la votación del aforo por parte de los usuarios finales.

---

---

# CAPÍTULO 4

---

## DEFINICIÓN DEL TRABAJO

En el presente capítulo se exponen los objetivos establecidos antes del inicio del desarrollo del proyecto (sección 4.1). Posteriormente, se detallará la metodología llevada a cabo a lo largo del proyecto (sección 4.2) y, finalmente, una estimación económica de los posibles costes incurridos así como la cronología empleada para el desarrollo de la aplicación (sección 4.3).

### 4.1. Objetivos

Tras una breve motivación y contextualización del proyecto en capítulos anteriores, en este apartado se van a detallar los diferentes objetivos del proyecto basado en la creación de esta aplicación.

#### 4.1.1. Gestión del aforo de las playas

El objetivo principal de este proyecto es el diseño y desarrollo de una aplicación que gestione el aforo de las playas en la isla de Ibiza. Con el desarrollo de la aplicación se permitirá visualizar el nivel de ocupación de las playas en tiempo real, basándose en la interacción de los usuarios que aportarán información sobre dicha playa. Cuanto mayor sea la participación ciudadana en la aplicación, más datos se acumularán en el sistema

y, por tanto, mayor será la precisión y fiabilidad del mismo.

La aplicación, dispondrá de una interfaz sencilla e intuitiva de forma que cualquier persona, de cualquier edad o país, sea capaz de consultar el estado de las diferentes playas y de contribuir al correcto funcionamiento de la aplicación estableciendo el aforo.

Además, se implementará un sistema de autenticación de usuarios, con roles diferenciados. Por un lado, existirán los administradores, que podrán llevar a cabo tareas como agregar, modificar o eliminar registros de playas, mientras que, por otro lado, los usuarios estándar podrán acceder únicamente a las funciones de consulta y participación.

### **4.1.2. Desarrollo de un algoritmo que prediga la hora de llenado de las playas**

La aplicación incorpora modelos de predicción basados en un algoritmo de *Machine Learning* con el fin de ayudar al usuario a planificar con antelación qué playa visitar en los siguientes días de su estancia. De esta forma, los usuarios podrán organizar mejor su tiempo en la isla, optimizándolo en todo momento.

Este modelo predice la hora aproximada en la que una playa alcanzará su nivel máximo de ocupación, teniendo en cuenta variables como las condiciones climatológicas, el día de la semana o la playa seleccionada.

### **4.1.3. Desarrollo de un asistente inteligente ‘Where to go’**

El proyecto también incluirá un sistema de recomendación con el fin de sugerir y aconsejar al usuario otras playas en caso de que la búsqueda esté llena. Estas recomendaciones se realizan en base a la cercanía a la playa original seleccionada y a sus características físicas como el tipo de arena entre otras.

Por ejemplo, si se desea visitar una playa con agua cristalina, arena fina y un kiosco, el sistema propondrá aquellas alternativas cercanas a la playa inicial que compartan el mayor número posible de estas características, facilitando así una elección informada y cómoda.

### 4.1.4. Fomento de un turismo sostenible

Finalmente, se pretende conseguir con esta aplicación una distribución más equilibrada del turismo en la isla y, facilitando la toma de decisiones basadas en la ocupación, también se busca minimizar los desplazamientos innecesarios, reducir la congestión y evitar aglomeraciones para así limitar el impacto que el turismo de masas tiene sobre el medioambiente y los ecosistemas de la isla.

## 4.2. Metodología

El desarrollo de este proyecto se ha llevado a cabo siguiendo una metodología mixta, combinando elementos del modelo *Waterfall*, donde cada etapa debe haberse completado antes de pasar a la siguiente, con prácticas de enfoques ágiles (*agile*), metodología basada en un enfoque iterativo e incremental, lo cual ha permitido adaptar el proceso a los cambios que han ido surgiendo durante su evolución.

Principalmente, se aplicó *waterfall* al inicio del proyecto, durante las primeras fases, con el fin de establecer una base sólida y funcional (especialmente con el diseño del *backend*) antes de comenzar con las funcionalidades de la aplicación. Además, se optó por una dinámica ágil más adelante, para iterar y mejorar funcionalidades básicas y simples. Esta última metodología favorece a la identificación temprana de errores, la incorporación de mejoras progresivas y una mayor flexibilidad ante imprevistos.

Para el desarrollo de la aplicación, se han seguido las siguientes fases:

- **Fase 1: Análisis y planificación.** En esta primera etapa se llevó a cabo la recopilación de requisitos, así como el diseño inicial de la arquitectura del sistema y la planificación del desarrollo.
- **Fase 2: Desarrollo del *backend*.** En esta fase se desarrolló el *backend* en Java utilizando el entorno de desarrollo *IntelliJ IDEA*. Se utilizó *Spring Boot* y se definieron entidades, controladores, repositorios, servicios y la configuración de seguridad.

- **Fase 3: Desarrollo del *fronted*.** Creación de la interfaz de usuario con tecnologías web (HTML, CSS y *Thymeleaf*), orientada a la simplicidad y usabilidad por parte del usuario final.
- **Fase 4: Pruebas.** Ejecución de diversas pruebas funcionales y de rendimiento para validar el correcto y esperado funcionamiento del sistema y detectar posibles errores.
- **Fase 5: Despliegue provisional.** Publicación inicial de la aplicación en un entorno de prueba para recopilar observaciones, comentarios y evaluar la experiencia de uso.
- **Fase 6: Evaluación y mejoras.** Análisis del *feedback* recibido, identificación de aspectos a mejorar y aplicación de los ajustes necesarios para mejorar y optimizar el sistema.
- **Fase 7: Despliegue definitivo y documentación.** Implementación de la versión final de la aplicación, así como la redacción de la memoria del proyecto, recogiendo todos los detalles técnicos.

Esta metodología híbrida ha ayudado a tener una estructura clara y definida desde el comienzo del proyecto, realizando a su vez un desarrollo más flexible y eficaz, enfocado en una mejora continua a partir de un resultado funcional sencillo.

### 4.3. Planificación y estimación económica

En este apartado se puede observar la planificación temporal que se ha llevado a cabo en cuanto al desarrollo de la aplicación (figura 4.1):

METODOLOGÍA DE TRABAJO	Noviembre	Diciembre	Enero	Febrero	Marzo	Abril	Mayo
Análisis y Planificación							
Desarrollo backend							
Desarrollo frontend							
Pruebas							
Despliegue							
Evaluación y mejoras							
Despliegue definitivo y elaboración de memoria							

Figura 4.1: Planificación temporal del trabajo. Fuente: Elaboración propia

#### 4. Definición del Trabajo

---

Además, se ha hecho una aproximación del coste que podría suponer el desarrollo de una aplicación de estas características (cuadro 4.1):

<b>Concepto</b>	<b>Coste estimado (€)</b>
Horas de desarrollo (150h x 25€/h)	3.750
API meteorológica profesional (plan básico anual)	100
Hosting y dominio (anual)	120
Mantenimiento técnico anual	600
<b>Total estimado</b>	<b>4.570</b>

Cuadro 4.1: Estimación de costes del proyecto. Fuente: Elaboración propia



---

---

## CAPÍTULO 5

---

# JUSTIFICACIÓN Y DESARROLLO DEL SISTEMA

En este capítulo se va a analizar en detalle el sistema implementado para el control y predicción del aforo en las playas de Ibiza. El proyecto se ha desarrollado siguiendo un enfoque modular, orientado a la escalabilidad y con una destacable interacción con el usuario. La aplicación combina tecnologías web, geolocalización, arquitectura REST y *Machine Learning*, integrando todo esto en un entorno de desarrollo seguro y adaptable.

Este sistema se ha diseñado bajo una estructura compuesta por distintos módulos (cliente, servidor, base de datos y modelo de predicción). Estos, aunque en esta versión corren sobre la misma máquina, han sido separados y claramente diferenciados para en un futuro poder ser desplegados de forma independiente. Así, se facilita el análisis de cada módulo, su mejora y evolución y, su implementación independiente.

Además, la figura 5.1 muestra un gráfico de la arquitectura del sistema para facilitar al lector su comprensión:

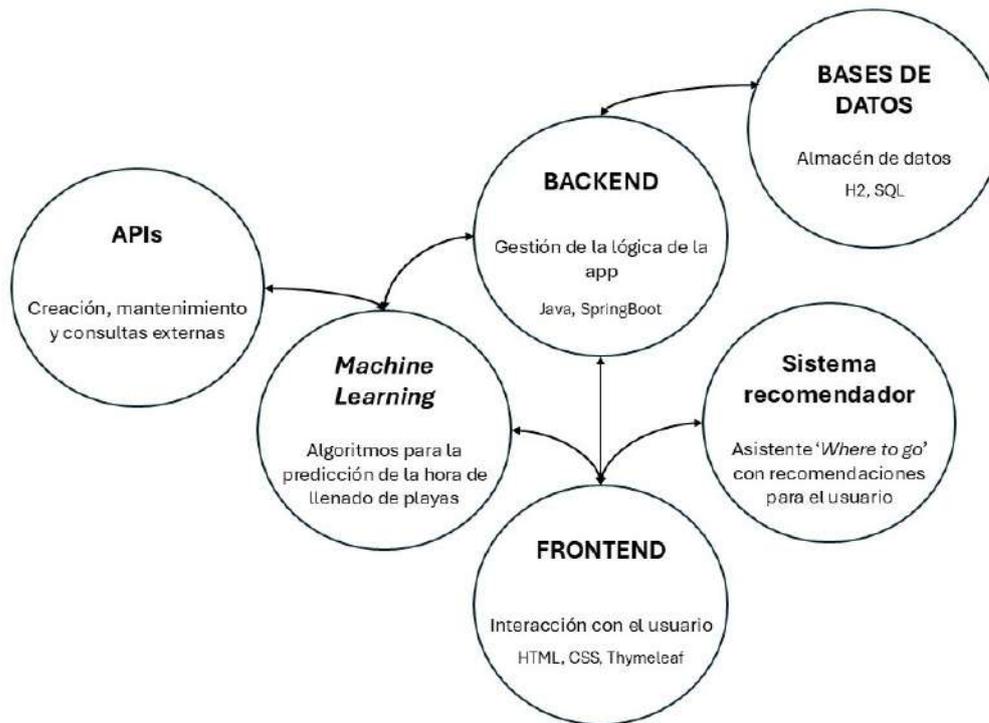


Figura 5.1: Arquitectura del sistema. Fuente: Elaboración propia

Este capítulo se divide en 2 secciones denominadas ‘análisis del sistema’, sección 5.1, donde se van a detallar los requisitos cumplidos en el proyecto y ‘diseño, implementación y justificación del sistema’, sección 5.2, donde se va a explicar el diseño del sistema, sus componentes principales y se van a justificar los rasgos destacados del proyecto.

## 5.1. Análisis del sistema

El proyecto, tal y como se ha repetido a lo largo del informe, busca gestionar de forma inteligente los flujos turísticos en las playas, concretamente en Ibiza. Para realizar esta gestión, se han creado un conjunto de funcionalidades claves que se detallan a continuación, dividiéndose en requisitos funcionales (subsección 5.1.1) y no funcionales (subsección 5.1.2).

### 5.1.1. Requisitos funcionales

Los requisitos funcionales son aquellos que el sistema necesita para poder satisfacer las expectativas y necesidades del usuario. En este caso son las siguientes:

### a) Consulta en tiempo real del aforo de las playas.

La aplicación permite a los usuarios consultar el nivel del aforo de una playa expresado en una de estas 5 palabras:

- *EMPTY*, significando que la playa está vacía.
- *LOW*, significando que está poco llena.
- *MEDIUM*, expresando que su capacidad se encuentra al 50 %.
- *HIGH*, que responde a una playa prácticamente llena pero que aún tiene algún espacio libre.
- *FULL*, que expresa que el aforo de la playa se encuentra al 100 % de su capacidad.

Esto varía según el tamaño de la playa, ya que una playa pequeña tal vez tiene un aforo máximo de *LOW* o *MEDIUM* a diferencia de una playa muy grande que su capacidad máxima será probablemente *FULL*. El aforo es establecido por los usuarios, funcionalidad que se detallará en el siguiente punto (sección 5.2) y, esta información se va actualizando de forma dinámica mediante una combinación de datos aportados por los propios usuarios. De esta forma, con esta funcionalidad, se puede observar inmediatamente el estado de las playas y así contribuir a una mejor distribución de aquellos que busquen visitar las zonas costeras ese día.

### b) Votación ciudadana sobre la ocupación percibida.

Los usuarios registrados pueden participar de forma activa en la mejora de la información del sistema, enviando su percepción sobre el nivel de aforo de una playa concreta. Respecto a como funciona el aforo de cada playa, ya se ha comentado previamente que el aforo máximo varía según el tamaño de la playa y esto afecta en las votaciones ya que no se podrá votar un aforo mayor al máximo establecido, ya que los niveles mostrados en el desplegable van hasta la capacidad máxima establecida. Los distintos votos se agregan a una base de datos, teniendo en cuenta el ID de usuario ya que no puede votar el mismo usuario más de una vez la misma playa en la misma hora.

El aforo se calcula haciendo una media del valor asociado a cada categoría (*Empty-0*,

*Low-1, Medium-2, High-3, Full-4*) en los votos de esa playa en la última hora. En el caso de no realizarse votos nuevos en los últimos 45 minutos, la media se realizará con los últimos tres votos realizados para esa playa y, este aforo, se reiniciará una vez llegadas las 19:00h UTC +01:00 para el día siguiente.

c) Registro, autenticación y roles diferenciados (usuario / administrador).

La aplicación cuenta con un sistema de autenticación y gestión de sesiones que permite el acceso a ciertas funcionalidades específicas según el tipo de usuario (usuario estándar o administrador). El mecanismo de inicio de sesión se basa en el uso de credenciales (correo electrónico y contraseña), gestionado desde el *backend* con verificación segura y persistencia de sesión utilizando *cookies* y *tokens*. En la base de datos de usuario se guarda el ID de usuario, el correo electrónico, la contraseña encriptada (seguridad de la cual hablaremos más adelante en este capítulo) y el tipo de usuario del que se trata.

Una vez se ha iniciado sesión, el sistema distingue entre el tipo de usuario y redirige al menú de inicio correspondiente que facilitará el acceso a las funcionalidades que le corresponden a cada uno:

- **Usuario estándar:** éste tiene acceso a las funciones básicas de la aplicación, entre las cuales encontramos la consulta de aforo de las zonas costeras, votación de la ocupación de una playa, visualización del mapa interactivo y utilización del modelo de *Machine Learning* para la predicción de la hora de llenado de las playas según datos meteorológicos.
- **Administrador:** además del acceso a las funcionalidades anteriormente mencionadas, el administrador tiene acceso a funcionalidades avanzadas que ofrecen herramientas específicas para añadir nuevas playas al sistema y, editar o eliminar playas existentes.

d) Visualización de características de cada playa (arena, agua cristalina, kiosco, etc.).

En el sistema, cada playa guardada en la base de datos consta de 5 características almacenadas como *booleans*: arena, que hace referencia a si la playa es de arena o de rocas; agua cristalina, que hace referencia a si el agua es color turquesa o si es más

oscura; kiosco, que indicará si la playa tiene un kiosco o restaurante; fumadores, que indica si se permite fumar en la playa; y mascotas, que informa de si la presencia de mascotas es permitida o no en la playa.

e) Sugerencia personalizada de playas menos saturadas.

En el sistema se puede apreciar que dentro de la funcionalidad de buscar una playa para observar su aforo también se observan sus características, descritas en el punto anterior. En el caso de que la playa consultada tuviese un aforo completo, se ofrecen recomendaciones sobre playas con el mismo número de coincidencias en cuanto a características, ordenadas en tablas de número de coincidencias y, en orden de cercanía a la playa consultada inicialmente.

f) Modelo de ML para predicción de la hora de llenado de las playas.

Este modelo de *Machine Learning* se detallará más adelante (subsección 5.2.5) de este mismo capítulo, pero ofrece al usuario una funcionalidad de consulta de la hora predicha de llenado de las playas. Es decir, el usuario selecciona la playa a la que le gustaría ir en los próximos días (al día siguiente, dentro de dos, o tres días) y el modelo devuelve la hora a la que calcula que llegará a su máxima capacidad basándose en factores climatológicos y en el día de la semana y el mes.

g) Ordenamiento de playas según proximidad a la ubicación del usuario.

Las coordenadas GPS proporcionadas por el navegador del usuario se utilizan, habiendo el usuario dado su previo consentimiento, para así mostrar el listado de todas las playas existentes en la base de datos ordenado por cercanía. Esto permite una experiencia personalizada y facilita la toma de decisiones rápidas, especialmente en contextos con limitaciones de tiempo o transporte.

h) Visualización de un mapa interactivo para la consulta más visual de las playas.

El proyecto contiene un mapa que ayuda a la visualización gráfica de las playas. En este mapa se puede previsualizar una imagen de la playa y, si se pincha sobre el nombre, se podrán apreciar todas las características. También, hay un icono de una

persona que se puede mover por toda la isla y que proporciona al usuario un listado con las playas que se encuentran en un radio aproximado de 5km con respecto a la ubicación del muñeco. De esta forma, se ayuda a los usuarios a que se ubiquen mejor haciendo uso de un mapa en lugar de ofrecerles únicamente los nombres de los municipios.

### 5.1.2. Requisitos no funcionales

Los requisitos no funcionales son aquellos que definen la calidad de un sistema y, a su vez, sus limitaciones como, por ejemplo, la seguridad o la escalabilidad. Algunos de estos conceptos ya se han mencionado a lo largo de la memoria y, ahora, se profundizará en ellos:

a) Interfaz web accesible desde cualquier dispositivo.

La aplicación está diseñada para ser *responsive*, es decir, adaptarse automáticamente a distintos tamaños de pantalla, permitiendo así su uso desde ordenadores, tabletas o móviles sin perder ningún aspecto de su funcionalidad ni degradar la experiencia del usuario. El diseño prioriza la usabilidad, con una estructura clara y navegación sencilla, para que todas aquellas personas que deseen usar la aplicación puedan hacerlo y que no vean en ella ninguna dificultad en su uso.

b) Sistema seguro con protección de rutas y control de sesión.

Se utiliza *Spring Security* como *framework* principal de protección en el *backend*, aplicando autenticación basada en *tokens* para controlar el acceso a funcionalidades restringidas. Las sesiones de usuario están protegidas y se utilizan buenas prácticas en el almacenamiento de contraseñas y gestión de roles. Las principales medidas de seguridad adoptadas son las siguientes:

- **Protección de rutas:** asegurando accesos diferenciados según el *endpoint* basado en roles.
- **Inicio de sesión con validación de credenciales:** mediante un sistema de *tokens* personalizados. Tras el *login*, se genera un *token* de sesión que se envía al navegador como una *cookie* segura hasta que el usuario cierra su sesión.

- **Control de acceso por rol (*Admin/User*):** acciones como la gestión de playas solo son accesibles para administradores autenticados. Estas validaciones están explícitamente detalladas en los controladores (*user.isAdmin()*).
- **Cierre de sesión y expiración de tokens:** se implementa una ruta para eliminar tokens y forzar el cierre de sesión con expiración inmediata de la *cookie*.
- **Cifrado de contraseñas:** con *BCryptPasswordEncoder* antes de almacenarlas en la base de datos. Esto permite una mayor seguridad ya que no son directamente legibles por el ser humano.  
Además, se emplea el uso de *salt* para obtener diferentes *hashes* aunque dos personas presenten la misma contraseña.
- **Validaciones adicionales:** en el *frontend* se realizan otras validaciones adicionales como la exigencia de unos requisitos estrictos para el campo de la contraseña en el formulario de registro (mayúscula, minúscula, número mínimo de caracteres, etc.) antes de permitir el envío de datos.
- **Restricción de CORS:** (por sus siglas en inglés *Cross-Origin Resource Sharing*) y el control de credenciales configurado para permitir el intercambio de *cookies* seguras entre el *backend* y el *frontend*.

c) Escalabilidad para incorporar más playas o ampliar funcionalidades.

La arquitectura modular permite la incorporación futura de nuevas playas, categorías de usuario o, incluso, otros destinos turísticos sin necesidad de rediseñar el sistema. De esta forma, si la aplicación quiere desplegarse en un sitio que no fuese la isla de Ibiza o si se desean añadir funcionalidades adicionales en un futuro se puede realizar de forma sencilla, sin necesidad de cambiar la estructura de la aplicación.

d) Datos en tiempo real mediante actualización dinámica.

Dada la información aportada por los usuarios y la constante actualización de la base de datos, se garantiza que los usuarios siempre tengan la última versión disponible sobre el aforo de la playa deseada. Esta información se actualiza sin necesidad de recargar la página web gracias al uso que se está haciendo al uso de peticiones

asíncronas (AJAX o llamadas REST).

e) Integración de servicios externos para enriquecer las funcionalidades.

El sistema se conecta a APIs externas, en la funcionalidad de calcular la hora de llenado de la playa, y así se consiguen obtener datos meteorológicos en tiempo real. Esto ayuda a ofrecer al usuario una experiencia realista y es necesario para poder ejecutar el modelo predictivo.

## 5.2. Diseño, Implementación y Justificación

Habiendo analizado las soluciones existentes al problema planteado (capítulo 3), se observa que hay una evidente carencia de soluciones completas que permitan controlar, predecir y gestionar el aforo de playas tal y como se plantea en esta aplicación.

Ninguna de las soluciones existentes combina una predicción basada en *Machine Learning* con un objetivo de obtener un turismo sostenible y de contribuir a una mejora medioambiental.

Por otra parte, este proyecto aporta la idea del uso de una interfaz que permita al usuario aportar datos, recogidos para así predecir de mejor forma la ocupación de las zonas costeras. Esta idea de interacción pretende dejar atrás la idea de una aplicación que informa unidireccionalmente tal y como se plantea en las ya existentes, convirtiendo al usuario en partícipe del proceso y agrandando su sentido de pertenencia. Esta solución favorecerá la optimización del uso del espacio público de forma más eficiente gracias a sus datos en tiempo real.

Con el fin de favorecer al medioambiente mediante el control de flujo turístico con un proyecto tan innovador, esta aplicación puede resultar de gran ayuda a administraciones públicas y organismos turísticos. Su implementación permitiría mejorar la gestión del turismo sin la necesidad de tener que desplegar nuevas instalaciones tecnológicas que sean económica y burocráticamente costosas y que puedan descartar proyectos necesarios e innovadores como de este tipo.

El sistema desarrollado se apoya en una arquitectura por capas, una estrategia ampliamente adoptada en el desarrollo de *software* por sus ventajas en organización, mantenibilidad y escalabilidad. Esta arquitectura permite dividir el sistema en módulos bien diferenciados que se encargan de tareas específicas, tales como la gestión de la interfaz de usuario, la lógica de negocio, el acceso a la base de datos, la interacción con servicios externos y la ejecución de algoritmos predictivos.

Cada una de estas capas se comunica con las demás a través de interfaces bien definidas, lo que garantiza independencia entre componentes y facilita futuras mejoras o cambios sin comprometer el resto del sistema. Esta separación de responsabilidades contribuye a la claridad del código, reduce el riesgo de errores y favorece el trabajo en equipo y la evolución del proyecto a largo plazo. También, cabe destacar que la escalabilidad del proyecto hace posible que, además de a playas, la aplicación pudiese extender su uso en un futuro a eventos al aire libre, parques naturales o cualquier entorno que pueda tener problemas de sobreocupación.

Los principales elementos en esta arquitectura son los siguientes (descritos en las siguientes subsecciones): *frontend*, *backend*, base de datos, servicios externos, sistema de predicción y seguridad.

Una vez definidos, se implementa el sistema utilizando las tecnologías correspondientes en cada capa y especificadas en el capítulo 2. El hecho de utilizar tecnologías modernas hace que la integración con otras herramientas o tecnologías sea sencilla. También, ayuda a eliminar configuraciones manuales y permite la creación de aplicaciones en poco tiempo. La estructura modular de estas tecnologías favorecen y contemplan un posible crecimiento de la aplicación en un futuro. A continuación, se van a detallar los elementos de la arquitectura, explicando así sus aspectos técnicos y prácticos.

### 5.2.1. Interfaz de usuario (*fronted*)

Este elemento constituye la capa visual del sistema, que ha sido desarrollada con tecnologías como HTML5 y CSS3, junto con *Thymeleaf*. Gracias a estas tecnologías ha sido posible la creación de una plantilla visual estructurada y estéticamente correcta. *Thymeleaf* ayuda en la integración con el *backend* y la generación dinámica del contenido

ya que se trata de un motor de plantillas especializado en proyectos de *Spring Boot*. El diseño de la interfaz es *responsive*, lo que significa que está preparado para un cambio de dispositivo sin perder ninguna de sus funcionalidades ni ningún detalle de la experiencia. Se podrá utilizar la aplicación tanto desde un ordenador como desde tabletas o móviles.

Entre las funcionalidades principales, accesibles desde el *frontend*, se incluyen:

- Consulta del nivel de ocupación en playas en tiempo real.
- Posibilidad de votar el aforo de la playa según se conciba.
- Visualización de características específicas de cada playa (arena, transparencia del agua, kiosco...).
- Recomendaciones personalizadas si el aforo de una playa buscada es completo.
- Predicción de la hora de llenado de la playa basada en datos meteorológicos.
- Gestión de perfil de usuario (para la edición de ciertos campos e incluso su borrado).
- Envío de un comentario o sugerencia a *S'AforApp* para dar *feedback* sobre una playa en concreto o en general.

Adicionalmente, los usuarios catalogados como administradores cuentan con secciones especiales dentro de la plataforma para realizar tareas como la gestión de playas (añadir, eliminar o editar playas).

La sencillez en el frontend es un aspecto clave para ofrecer una buena experiencia de usuario (UX). Muchas aplicaciones fracasan en este sentido por ser poco intuitivas o difíciles de utilizar, lo que aleja a los usuarios. En cambio, una interfaz sencilla, intuitiva y fácil de utilizar ayuda a que personas de todas las edades la puedan utilizar sin complicaciones, permitiendo un uso más constante.

### 5.2.2. Servidor (*backend*)

Esta capa se trata del núcleo de la lógica del sistema, el *backend*. Éste está desarrollado en Java mediante *Spring Boot*. Esta tecnología, tal y como se ha detallado en el

capítulo 2, ha sido elegida por su modularidad, robustez y excelente integración con herramientas modernas de desarrollo web.

Un *backend* modular y seguro es muy recomendable en un proyecto tanto por la mejora del sistema en cuanto a rendimiento como por la mejora en cuanto a escalabilidad, mantenibilidad y protección. El concepto de *backend* modular hace referencia a una arquitectura donde las funcionalidades se dividen en módulos, favoreciendo la localización más fácilmente de errores, la realización de mejoras, el trabajo en equipo y la reutilización del código.

Por otra parte, un *backend* seguro, ayuda a proteger los datos de los usuarios y evita el acceso de atacantes al sistema.

El *backend* tiene como principal función, recibir y procesar las peticiones provenientes del *frontend*. También aplica la lógica correspondiente, gestiona los datos y coordina la interacción con servicios externos.

Dentro del *backend* se encuentran los siguientes módulos clave:

- Módulo de autenticación y gestión de sesiones con *Spring Security*.
- Módulo de usuarios, roles y permisos.
- Control de votaciones, gestión de playas y almacenamiento de aforos.
- Comunicación con la API de predicción desarrollada en *Python*.
- Integración con APIs externas para obtener datos meteorológicos.
- Validaciones y tratamiento de errores mediante *Jakarta Validation*.

La lógica está organizada con *endpoints* REST estructurados que permiten su potencial uso desde aplicaciones móviles u otras plataformas futuras.

### 5.2.2.1. Funciones principales

Haciendo uso de los módulos previamente nombrados y de los *endpoints* REST se han conseguido llevar a cabo las siguientes funciones principales:

- Consultar el listado de playas y los datos actuales de aforo.
- Almacenar votaciones sobre la ocupación percibida por los usuarios.

- Acceder a recomendaciones personalizadas basadas en predicciones.
- Registrar nuevos usuarios, iniciar sesión y gestionar roles (usuario o administrador).
- Realizar tareas de administración como alta, edición o eliminación de playas.

### 5.2.2.2. Tecnologías utilizadas

Para el correcto funcionamiento del *backend*, se han empleado varios módulos de *Spring Boot* que se explican a continuación:

- **spring-boot-starter-web**: para manejar las peticiones HTTP y exponer *end-points* REST.
- **spring-boot-starter-data-jpa**: para trabajar con la base de datos usando JPA e *Hibernate*.
- **spring-boot-starter-security**: para gestionar la seguridad de la aplicación, incluyendo la autenticación y autorización.
- **spring-boot-starter-thymeleaf**: para generar páginas HTML dinámicas desde el servidor.

### 5.2.3. Base de datos

Durante la fase de desarrollo se ha optado por una base de datos H2<sup>1</sup> en memoria, que permite trabajar con rapidez sin necesidad de configuración adicional. Se emplea JPA junto con *Hibernate*, que facilita la manipulación de información directamente desde el código Java. También destaca por su utilidad a la hora de hacer pruebas.

Las principales tablas o entidades del modelo son:

- **Usuarios**: con su información personal (nombre y correo electrónico), contraseña encriptada y nivel de acceso (usuario/administrador).
- **Playas**: con nombre, datos geográficos, coordenadas y características.
- **Votaciones**: playa para la que se ha realizado el voto, registros de percepciones de ocupación aportadas por los usuarios, con la hora en la que se ha realizado la

---

<sup>1</sup>Información detallada sobre H2

votación y el ID del usuario que ha realizado la votación.

- **Feedback:** comentario realizado, playa para la que se ha realizado, email, nombre y teléfono (opcional) de la persona que deja el comentario, tipo de comentario, fecha de visita e ID de la playa en caso de que el comentario sea sobre una playa y opción de si el usuario desea recibir respuesta por parte de *S'AforApp* a su comentario.
- **Token:** con el ID de sesión y el ID del usuario que ha iniciado sesión.

A continuación se muestran unos ejemplos de las tablas de la base de datos descritas anteriormente.

La figura 5.2 muestra un ejemplo de la tabla de Usuarios de la base de datos, donde se puede ver la encriptación de las contraseñas para una mayor seguridad. Se ha empleado el uso de *salt*, valor aleatorio que se añade a una contraseña antes de cifrarla para dificultar su obtención y que cada *hash* sea único, incluso si dos usuarios poseen la misma contraseña:

SELECT * FROM USUARIOS;				
ID	EMAIL	NAME	PASSWORD	ROLE
12	adriana@admin.com	AdrianaAdmin	\$2a\$10\$yHxgrt80gb7PdPLcFjAnOfLrRR2z4gOHR7hYya9ME9V7daPU4wpO	1
13	adrianatm@admin.com	AdrianaTM	\$2a\$10\$oo4aZ2DmqBijAviC7VIF2uKVZcBbv38SsQ.IoFuQOT5URF1If9Ovq	1
14	luis@admin.com	LuisAdmin	\$2a\$10\$q3nPbb.Ie7Pv6DKX3xF.w.cUNp4vzBfsZQfOg2ElqBXyj/6f5Osq	1
16	adriana@user.com	AdrianaUser	\$2a\$10\$FTI4MwIEHsvtkQBjUsFqUuwxW3Ly63wMkX5MwqP.GRATrmhybPuz6	0
17	natividad@mari.com	Natividad	\$2a\$10\$BtCNbHE0Mv9w8.7H80ppguZQ.eIQcVWMHv-Ha7Ah1nIRUPuCqm7hDu	1
18	luis@torres.com	Luis Torres	\$2a\$10\$8b6VhybP/CmJKrGcQKKnehaeG.NPh7qbYKaCMKamHFI0GP.JqUeHe	1
20	atilano@comillas.com	Atilano	\$2a\$10\$nMs7CJ4IQzvkW4S9DOsouGtQaZB3FZwAnik1Ud5bBYAmo0YQeglm	1

Figura 5.2: Tabla de Usuarios, base de datos. Fuente: Elaboración propia

La figura 5.3 muestra un ejemplo de la tabla de Playas de la base de datos, donde se recoge la capacidad actual de las playas, sus nombres, coordenadas y características propias entre otros atributos:

SELECT * FROM BEACH;													
ID	ACTUAL_CAPACITY	BEACHNAME	LAT	LNG	IS_FULL	MAXIMUM_CAPACITY	MUNICIPIO	AGUA_CRISTALINA	ARENA	KIOSKO	FUMADORES	MASCOTAS	IMAGE_PATH
1	4	Algües Blanques	39.0570529	1.5813294	TRUE	4	0	TRUE	TRUE	TRUE	FALSE	TRUE	algues_blanques.jpg
2	0	Benirràs	39.0892175	1.4362218	FALSE	3	1	TRUE	TRUE	TRUE	TRUE	FALSE	benirras.jpg
3	3	Cala Bassa	38.9663884	1.235133	FALSE	4	4	TRUE	TRUE	TRUE	FALSE	FALSE	cala_bassa.jpg
4	1	Cala Boix	39.0434028	1.4940065	FALSE	3	0	FALSE	TRUE	TRUE	FALSE	TRUE	cala_boix.jpg
5	1	Cala Carbó	38.894362	1.2135247	FALSE	2	4	TRUE	FALSE	TRUE	TRUE	FALSE	cala_carbo.jpg
6	1	Cala Codolar	38.9474449	1.2201134	FALSE	2	4	FALSE	FALSE	FALSE	TRUE	FALSE	cala_codolar.jpg
7	3	Cala Comte	38.9625032	1.2201411	FALSE	4	4	TRUE	TRUE	TRUE	FALSE	TRUE	cala_comte.jpg

Figura 5.3: Tabla de Playas, base de datos. Fuente: Elaboración propia

## 5. Justificación y Desarrollo del Sistema

La figura 5.4 muestra un ejemplo de la tabla de *Votaciones* de la base de datos, donde se registra el valor resultante de la votación, el *timestamp* de la votación, el id de la playa en la que se ha votado y se guarda también el id del usuario que ha realizado dicha votación:

33	EMPTY	2025-05-26 05:27:58.100756	7	<i>null</i>
34	FULL	2025-05-26 05:28:28.467399	7	<i>null</i>
35	HIGH	2025-05-26 05:41:25.176684	1	<i>null</i>
36	FULL	2025-05-26 05:41:37.536377	1	<i>null</i>
37	FULL	2025-05-26 06:35:52.776016	7	<i>null</i>
38	LOW	2025-05-26 07:00:38.114894	5	12
39	LOW	2025-05-26 07:01:11.395193	6	12

Figura 5.4: Tabla de *Votaciones*, base de datos. Fuente: Elaboración propia

La figura 5.5 muestra un ejemplo de la tabla de *Feedback* de la base de datos en la que se recoge la información del usuario (nombre, email y número de teléfono - opcional), información de la playa sobre la que se deja el comentario, en caso de ser sobre una playa (ID, fecha de visita), comentario, tipo de comentario y si se desea recibir respuesta o no:

SELECT * FROM FEEDBACK;								
ID	COMENTARIO	EMAIL	FECHA_VISITA	NOMBRE	PLAYA	RESPUESTA	TELEFONO	TIPO_COMENTARIO
1	Todos los restaurantes están cerrados.	adr_dad@grupotisa.com	2025-06-04	Adriana	17	si		critica
2	Una playa preciosa.	adr_dad@grupotisa.com	2025-06-04	Adriana	18	si		positivo
3	Una cala preciosa.	adr_dad@grupotisa.com	2025-06-03	Adriana	18	si		positivo
4	Más plazas de parking.	adr_dad@grupotisa.com	2025-06-03	Adriana	12	si		sugerencia
5	Más plazas de parking.	adr_dad@grupotisa.com	2025-06-03	Adriana	21	si		critica
6	La App me gusta mucho.	luistorresserrano00@gmail.com	2025-06-10	Luis	ninguna	si		positivo
7	Una aplicación muy útil.	luistorresserrano00@gmail.com	<i>null</i>	Luis	ninguna	no		positivo
8	¡Preciosa playa!	jetur@grupotisa.com	2025-05-28	Juan Enrique	14	no		positivo
9	Ampliar el parking.	jetur@grupotisa.com	2025-05-28	Juan Enrique	16	no		sugerencia

Figura 5.5: Tabla de *Feedback*, base de datos. Fuente: Elaboración propia

La figura 5.6 muestra un ejemplo de la tabla de *Token* de la base de datos, que almacena tanto el ID del usuario que ha iniciado sesión como el ID de la sesión asociada a dicho usuario en un particular momento:

```
SELECT * FROM TOKEN;
```

ID	BEACH_ID	USUARIO_ID
2e961ce9-fc72-4ea4-8ab0-467ff5e12e36	null	13
494d90dd-4948-47b7-973a-ed406954dbe2	null	14
f9b1aa5f-714c-4f56-a0aa-25eb1677a3c4	null	17
6a183152-5bd7-483b-9643-1d03e0a9205b	null	18
9a5e29ca-d4c1-40f7-ab4a-ce6f04cc1aad	null	16
1c5b4f6b-7b8e-41b3-b82a-e373ca7d2987	null	20
97f34878-2064-4860-8eb8-e9c54ccc5127	null	12

Figura 5.6: Tabla de *Token*, base de datos. Fuente: Elaboración propia

Gracias al uso de *Hibernate*, la aplicación podría adaptarse fácilmente en un futuro a bases de datos más robustas cuando la aplicación se ponga en un entorno real en producción, como *MySQL* o *PostgreSQL*.

Finalmente, se muestra a continuación un diagrama entidad-relación en la figura 5.7, lo que permite estudiar la relación entre las distintas tablas explicadas de la base de datos:

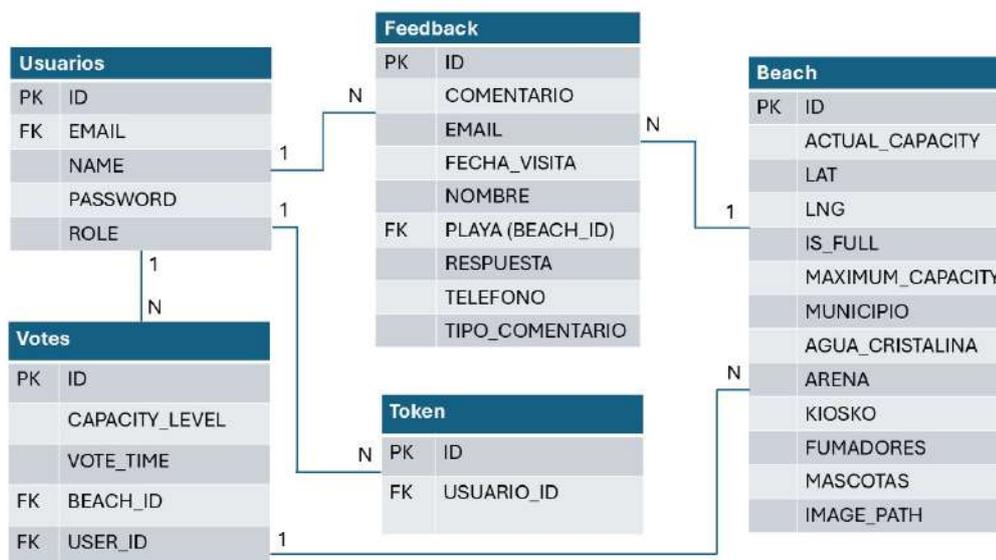


Figura 5.7: Diagrama entidad-relación, base de datos. Fuente: Elaboración propia

### 5.2.4. Servicios meteorológicos externos

Para conocer el estado actual del clima en la isla de Ibiza, la aplicación se conecta a servicios meteorológicos de terceros, es decir, a APIs externas.

Una de ellas, como se describió en la sección 2.10, sirve para obtener el horario de la salida y puesta de sol (valores con los que se puede calcular el número total de horas de ese día) y, la otra, para obtener datos sobre el estado meteorológico de la isla en los próximos días (temperaturas, nubosidad, presión, viento...).

Esta información tiene un doble objetivo:

- Por un lado, se le muestra al usuario para darle información sobre el día que desee a modo consultivo.
- Por otro lado, estos datos se le pasan al modelo de ML como valores de entrada para que pueda devolver la hora de predicción.

### 5.2.5. Sistemas predictivos (*Machine Learning*)

Uno de los componentes más innovadores del proyecto, en comparación con el resto de aplicaciones con un fin similar, es la predicción de la hora de ocupación total de las playas. Ninguna de ellas cuenta con algoritmos que predigan el aforo o conceptos similares.

Hoy en día, con la gran abundancia de datos que se generan, la IA está en auge y, especialmente, los modelos de *Machine* y *Deep Learning* que puedan obtener patrones de datos pasados y que hagan predicciones seguras que favorezcan la experiencia de los visitantes.

Para cada playa, se ha elaborado un modelo entrenado con datos históricos (la idea es que en el futuro estos datos se completen con los que se van introduciendo en la aplicación por parte de los usuarios) y cuyas variables de entrada son:

- El nombre de la playa seleccionada para hacer la predicción. No es una variable en sí del modelo pero es necesario para identificar qué modelo correr con la selección del usuario en el *frontend* y los datos obtenidos de las APIs.
- El día de la semana, como un valor entero entre 0 y 6 que luego se convierte

en variables *dummies* (variables binarias -0 ó 1- que se utilizan para representar categorías cualitativas).

- El mes del año, como un valor entre 1 y 12. También se convierte a variables *dummies*.
- Las horas de sol para ese día, en formato decimal, calculadas como la resta entre la hora de la puesta y la salida del sol en Ibiza.
- La temperatura máxima en grados.
- La temperatura mínima en grados.
- La intensidad del viento en km/h.
- La dirección del viento en coordenadas  $x$  e  $y$ . Para ello, se han tenido que calcular transformando los grados obtenidos de la API a radianes y luego empleando las funciones seno y coseno para una mejor representación de las mismas.
- Porcentaje de humedad.
- Porcentaje de nubosidad.
- Presión atmosférica en milibares.
- Y la variable de salida es la hora predicha de llenado, en formato decimal, para esa playa bajo las condiciones indicadas.

Para entrenar los modelos se han realizado las siguiente acciones:

- Separar en dos conjuntos diferentes y en una proporción 80 %-20 % el conjunto de datos históricos para entrenar el modelo y luego testarlo con datos nunca vistos por él.
- Probar diferentes modelos y algoritmos como *RandomForestRegressor* o *XGBRegressor* (de la librería *XGBoost*) aunque finalmente se optó por ***GradientBoostingRegressor***<sup>2</sup> de la librería *sklearn* por su, aunque ligero, mejor rendimiento (cuadro 5.1).

---

<sup>2</sup>Documentación oficial del *GradientBoostingRegressor* en *sklearn*

Algoritmo	R <sup>2</sup> medio
<i>GradientBoostingRegressor</i>	70,04
<i>XGBRegressor</i>	70,02
<i>RandomForestRegressor</i>	58,81

Cuadro 5.1: Métricas de los mejores algoritmos. Fuente: Elaboración propia

- Emplear técnicas de Validación Cruzada. Con ello, se consigue evaluar el rendimiento de un modelo dividiendo los datos en varias partes, entrenándolo en unas y validándolo en otras, para asegurar que generaliza bien a datos no vistos. Se ha empleado *GridSearchCV* con 5 carpetas, ya que además permite la optimización de los hiperparámetros de cada modelo de forma automática (por ejemplo, *n\_estimators*, *learning\_rate* o *max\_depth*).
- Calcular algunas métricas a la hora de testear el modelo con el conjunto de *test* para evaluar su rendimiento. Entre ellas, destacan:
  - R<sup>2</sup>: valor que mide qué proporción de la variabilidad de los datos explica el modelo. Cuanto más cercano a uno, mayor fiabilidad tendrá el modelo. Se calcula con la siguiente ecuación (5.1):

$$R^2 = 1 - \frac{\sum_{i=1}^n (y_i - \hat{y}_i)^2}{\sum_{i=1}^n (y_i - \bar{y})^2} \quad (5.1)$$

- MSE, Mean Squared Error o Error Cuadrático Medio: calcula el error cuadrático medio penalizando más los errores grandes. Se calcula con la siguiente ecuación (5.2):

$$\text{MSE} = \frac{1}{n} \sum_{i=1}^n (y_i - \hat{y}_i)^2 \quad (5.2)$$

- MAE, Mean Absolute Error o Error Absoluto Medio: mide el error absoluto medio, indicando el promedio de las diferencias absolutas entre predicciones y valores reales. Se calcula con la siguiente ecuación (5.3):

$$\text{MAE} = \frac{1}{n} \sum_{i=1}^n |y_i - \hat{y}_i| \quad (5.3)$$

Se han calculado dichas métricas para cada modelo de *GradientBoostingRegressor* variando los hiperparámetros, obteniendo diversos resultados en función de la playa y de sus datos históricos. El  $R^2$  medio de los modelos se sitúa en torno a 0.70 y en el cuadro 5.2 se presentan algunos de los que han tenido mejor rendimiento en base al  $R^2$ .

<b>Playa</b>	<b>MSE</b>	<b>MAE</b>	<b>R<sup>2</sup></b>
Cala Gració	0.1045	0.26 horas	0.8589
Es Torrent	0.0667	0.21 horas	0.8356
Cala d'Hort	0.0845	0.22 horas	0.8343
Es Canaret	0.0936	0.26 horas	0.8314
Pou des Lleó	0.0990	0.27 horas	0.8264
Cala Pada	0.1285	0.29 horas	0.8057
Cala Xarraca	0.1109	0.24 horas	0.8034

Cuadro 5.2: Métricas de los mejores modelos de predicción. Fuente: Elaboración propia

Como se aprecia, el MAE da valores de aproximadamente 15-20 minutos de error, lo cual da una idea de la utilidad y buen funcionamiento de los modelos.

Posteriormente, cada uno de estos modelos ya entrenados se han guardado en archivos .pkl y se han integrado en la API desarrollada, de forma que la comunicación dentro de la aplicación es muy sencilla. El usuario escoge una playa y un día que quiera predecir, los datos se envían al modelo entrenado para dicha playa seleccionada y éste devuelve una hora de predicción que se le transmite al usuario de vuelta.

### 5.2.6. Seguridad

La seguridad es un componente esencial en cualquier aplicación que gestione datos personales y niveles de acceso diferenciados. Para ello, se ha incorporado *Spring Security*, un módulo especializado en proteger rutas sensibles y garantizar que los usuarios solo puedan realizar acciones autorizadas.

Entre las funcionalidades de seguridad principales destacan:

- Control de acceso por roles (usuario o administrador).
- Protección de *endpoints* sensibles contra accesos no autorizados.
- Validación de credenciales durante el inicio de sesión.

- Gestión segura del cierre de sesión.

### 5.2.7. Diagrama de componentes

La figura 5.8 muestra el diagrama de componentes de la aplicación al completo:

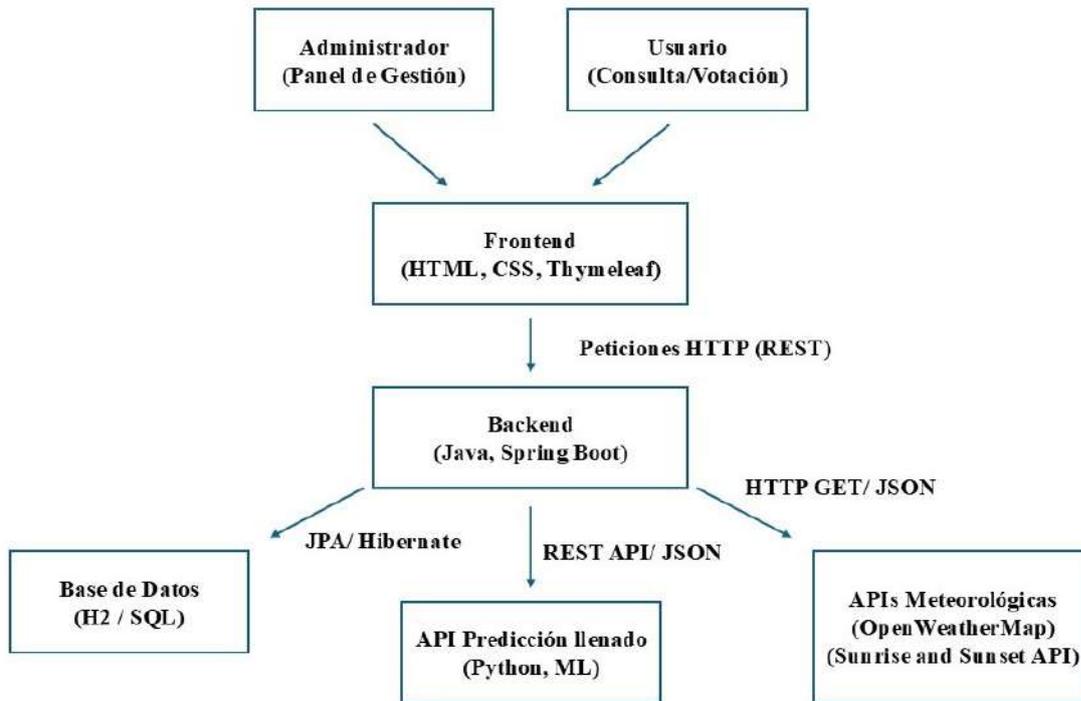


Figura 5.8: Diagrama de componentes. Fuente: Elaboración propia

### 5.2.8. Justificación práctica y social

Una vez detalladas y justificadas las tecnologías, se comenta como esta aplicación también tiene un impacto positivo en el ámbito social:

#### 1. Mejora de la experiencia por parte del turista.

Con esta aplicación, la experiencia del turista mejorará considerablemente por el simple hecho de poder planear sus vacaciones. Poder conocer playas nuevas que tal vez no son las más populares pero siguen siendo igual de bonitas es una opción novedosa para el turista y, el poder saber cómo de llena se encuentra esa playa todavía más. Dando así la opción de elegir basándose en datos.

### 2. **Mejora de la vida cotidiana del residente.**

El residente mejora su vida cotidiana dado que tiene la opción de saber si la playa a la que quiere asistir está llena y a que hora se realizará su llenado. Muchas veces hay playas a las que los residentes de la isla no intentan ir porque dan por hecho que una vez lleguen ya no habrá espacio de forma que esta aplicación da la opción de poder volver a esas famosas playas, anticipándose a su hora de llenado y viendo las opciones que tienen.

### 3. **Ayuda al medioambiente al reducir desplazamientos innecesarios por alta ocupación en las zonas costeras.**

Esta es una de las principales razones por las que esta aplicación supone un cambio. Son muchas las veces que se llega a una playa y ya no se permite el paso porque su aforo es completo y, hay que redirigirse a otra playa haciendo un uso excesivo del vehículo y creando problemas de tráfico y contaminación. De esta forma, se podría saber seguro en qué playa hay aforo y así anticiparse y evitar este problema.

### 4. **Posible ayuda a la gestión municipal o insular para políticas de acceso, planificación o conservación.**

Esta mejora ayuda considerablemente a la gestión municipal ya que permite saber la distribución del turismo y poder así realizar una gestión más eficiente por parte de los ayuntamientos u otros organismos, con el fin de coordinar mejor el flujo de personas en la isla.

## 5.2.9. **Diferencias frente al mercado actual**

Tal y como se ha expresado previamente, hoy en día no existe ninguna aplicación que combine todas las características y funcionalidades descritas en *S'AforApp*.

*iPlaya* es una aplicación con un fin informativo, careciendo de interacciones; *PlayasApp* posee un enfoque distinto dada la necesaria instalación de cámaras para su funcionamiento, lo cual limita su funcionalidad a aquello que enfoquen las cámaras; o, *Platges Segures*, que supone una solución temporal a un problema, en aquel momento, la pan-

demia del Covid-19.

En cambio, a diferencia de todas las soluciones mencionadas, la propuesta supone una solución adaptable a cualquier destino y con un enfoque escalable, lo cual aporta un margen para mejoras y nuevas funcionalidades.

### **5.2.9.1. Turismo sostenible**

Además de ofrecer diferencias respecto a otros proyectos, este también se alinea con los Objetivos de Desarrollo Sostenible (ODS), los cuales se detallarán en el anexo A. De esta forma, al distribuir más el flujo de turistas en las playas, se contribuye a reducir el impacto que tiene el turismo en el medioambiente.

---

---

# CAPÍTULO 6

---

## ANÁLISIS DE RESULTADOS

Dado que este proyecto se ha centrado en el desarrollo de una aplicación y no se han obtenido resultados como tal, el análisis de resultados se va a centrar en la verificación del cumplimiento de los objetivos establecidos inicialmente y detallados en el capítulo 4. A continuación, se procede a detallar los logros alcanzados en relación con dichos objetivos.

Por su parte, el anexo B, presenta un manual de uso de la aplicación para los usuarios que quieran conocer todas las funcionalidades de la *app* como resultado del desarrollo del proyecto.

### **6.1. Objetivo 1: Gestión del aforo de las playas**

El objetivo principal consistía en el diseño de una aplicación para gestionar el aforo de las playas en Ibiza, ofreciendo información a tiempo real y de esta forma mejorar la experiencia diaria en la isla. Este objetivo se ha conseguido con el desarrollo de una aplicación funcional que permite tomar decisiones de forma más fácil teniendo información sobre el aforo de las playas en cada instante.

Este desarrollo se ha llevado a cabo aplicación con una interfaz sencilla e intuitiva, mostrando el nivel de ocupación de las playas en tiempo real gracias a la interacción del

usuario, que proporciona información sobre el aforo de las playas en base a cómo este lo perciba. Esta, entre otras funcionalidades, forman esta aplicación y, por lo tanto, se puede considerar que este objetivo se ha cumplido satisfactoriamente.

## **6.2. Objetivo 2: Algoritmo para la predicción de la hora de llenado de las playas**

Se ha diseñado e implementado un modelo de *Machine Learning* que estima la hora de llenado de las playas en función de parámetros relacionados con el clima o el día de la semana. De esta forma, el usuario podrá ver a qué hora se prevé que se llenará la playa a la que quiere acudir para así calcular bien su hora de llenado o elegir otra playa.

## **6.3. Objetivo 3: Desarrollo del asistente inteligente ‘Where to go’**

“Where to go” es una funcionalidad desarrollada con el fin de aconsejar al usuario sobre a qué playa debe asistir si la buscada está llena. Esto se ha conseguido integrando un sistema recomendador que, identificando las características de la playa buscada, recomienda aquellas playas más similares en orden de cercanía a la playa buscada.

## **6.4. Objetivo 4: Fomento de un turismo sostenible**

A través de la creación de esta aplicación, de la información que se proporciona en tiempo real y de las recomendaciones personalizadas, se favorece a una mejor distribución de los visitantes en las playas de la isla. Esto permite que haya un equilibrio y no se saturen ciertas zonas y evita desplazamientos innecesarios. De esta forma, se puede confirmar que se fomenta un turismo sostenible logrando así el cumplimiento de este tercer objetivo.

Habiendo detallado todos los objetivos y cómo estos se han alcanzado a lo largo del

desarrollo del proyecto, se puede concluir que los resultados han sido acordes a lo esperado. La aplicación implementada cumple con los requisitos funcionales planteados inicialmente y presenta un nivel técnico suficiente para ser considerada una solución viable dentro del ámbito para el que fue diseñada.

Además, se ha demostrado que es posible abordar una problemática real mediante el uso de herramientas tecnológicas actuales, aplicando conceptos tanto de desarrollo de *software* como de Inteligencia Artificial. En conjunto, el proyecto no solo ha permitido alcanzar los objetivos definidos, sino que también ha aportado valor práctico y una base sólida para futuros desarrollos y mejoras e, incluso, una implementación real.



---

---

# CAPÍTULO 7

---

## CONCLUSIONES Y TRABAJOS FUTUROS

Este capítulo tiene como finalidad hacer una reflexión final sobre el desarrollo del proyecto indicando el trabajo realizado, los logros alcanzados y, a su vez, indicando posibles líneas de trabajo que puedan mejorar o ampliar el proyecto desarrollado. De esta forma, se deja constancia del estado actual, abriendo la puerta a futuras mejoras que refuercen su utilidad y viabilidad.

Este capítulo se organiza en 2 secciones denominadas conclusiones, sección 7.1, donde se va a resumir el trabajo realizado y a reflexionar sobre aquello en lo que se ha trabajado y aprendido y, trabajos futuros, sección 7.2, donde se van a detallar las distintas vías de mejora y próximos pasos que se ofrecen para el futuro de la aplicación.

### **7.1. Conclusiones**

El desarrollo de esta aplicación ha supuesto una experiencia enriquecedora y fructífera que ha favorecido a mi desarrollo tanto técnico como conceptual. Me ha dado la oportunidad de aplicar mis conocimientos en distintos campos como el diseño, la programación

e incluso la Inteligencia Artificial, ofreciendo así una solución al que es un problema real en mi hogar, la isla de Ibiza.

Durante estos meses, he logrado construir una solución a la problemática planteada al principio: mejorar la gestión del aforo en las zonas costeras de Ibiza mediante el uso de tecnologías accesibles. La aplicación se encuentra operativa y abierta a seguir evolucionando y mejorando con el fin de ofrecer a los usuarios siempre la última y la mejor de las versiones.

Técnicamente, se ha diseñado una interfaz amigable e intuitiva, un sistema de autenticación por roles, ofreciendo distintas funcionalidades para usuarios y administradores y, un módulo inteligente capaz de recomendar playas y anticipar su ocupación. También, se ha trabajado con criterios de simplicidad con el fin de que no se necesiten grandes conocimientos tecnológicos para utilizar esta aplicación.

Esta aplicación me ha ofrecido la oportunidad de estrenarme en un aspecto nuevo para mí, que es el *Machine Learning*, aprendiendo y formándome sobre él y así poder llegar a desarrollar un modelo básico de predicción.

En definitiva, este proyecto representa una aportación funcional y actual que responde a una necesidad real y plantea una alternativa digital, sostenible y con visión de futuro.

### 7.2. Trabajos futuros

De cara a la evolución del proyecto, existen distintos aspectos en los que se podría mejorar o vías por las cuales se podrían ampliar sus funcionalidades. Todas estas se van a detallar a continuación:

- Aumento de la automatización.

En este aspecto, la aplicación podría mejorar instalando sistemas tecnológicos que favorezcan al funcionamiento de la aplicación como sensores físicos, cámaras o APIs de datos abiertos. De esta forma la recopilación de datos sería más fiable y no se basaría simplemente en la opinión de los usuarios.

- Optimización del modelo de *Machine Learning*.

Otra de las mejoras pensadas para la aplicación era la optimización y mejora del modelo de *Machine Learning*. Esto podía realizarse uniendo la tabla con la que se entrena el modelo a la tabla de votos de la base de datos y, de esta forma, con un aumento de datos históricos y con datos más precisos sobre el estado de aforo de cada playa, que el modelo de predicción proporcione resultados más acertados.

Además, otra posible mejora sería el uso de distintos algoritmos (y no sólo modelos) en función de la playa.

- Expansión de sus fronteras.

La internacionalización del proyecto o el simple hecho de expandir sus fronteras, no es algo complicado y puede ayudar a mejorar el negocio y a dar a conocer la aplicación en otros destinos costeros. También cabe la posibilidad de extender estas funcionalidades a parques, lagos u otras zonas públicas en las que se pueda encontrar una gran aglomeración de personas.

- Clasificación de usuarios según su comportamiento en la aplicación.

Con el objetivo de incentivar un uso responsable y frecuente de la herramienta, se podría implementar un sistema de reconocimiento que otorgue mayor peso a las votaciones realizadas por usuarios fiables. Asimismo, se podrían ofrecer recompensas en forma de funcionalidades adicionales o distintivos visibles dentro de la aplicación que reflejen su compromiso.

- Colaboración con entidades públicas y recompensas al usuario.

Para incentivar la colaboración de los usuarios, se podría recompensar con bonos para gastar en establecimientos del municipio o de la isla. Esto podría ser posible si se colaborase con ayuntamientos o entidades públicas. Esta colaboración, también podría llevar a la implantación real de la aplicación como servicio para tanto turistas como residentes y así ofrecer una vía de organización funcional y común para todos.

En resumen, este proyecto ha permitido materializar una idea con un importante impacto en la gestión turística de la isla de Ibiza. Haciendo uso de la tecnología se ha desarrollado una herramienta útil, accesible y sostenible. La experiencia ha sido valiosa

tanto a nivel personal como técnico y, aunque es cierto que todavía existen mejoras por desarrollar y existen posibilidades de expansión, a día de hoy, se consta de una base sólida y utilizable. De esta forma se concluye que esta aplicación refleja un nuevo enfoque de las herramientas tecnológicas y, en este caso, una solución novedosa dada su interacción con el usuario, que puede suponer un antes y un después en la gestión turística.



---

---

# APÉNDICE A

---

## ALINEACIÓN DEL PROYECTO CON LOS ODS

Los Objetivos de Desarrollo Sostenible (ODS) forman parte de la Agenda 2030 de las Naciones Unidas, un plan de acción global adoptado por todos los Estados miembros para erradicar la pobreza, proteger el planeta y garantizar la prosperidad para todas las personas. Cada uno de los 17 objetivos plantea metas específicas que los países deben alcanzar de aquí a 2030 y, en los últimos años, se ha impulsado especialmente el papel de la tecnología como herramienta clave para acelerar su cumplimiento.

En el contexto de este proyecto, centrado en la creación de una aplicación para la gestión inteligente del aforo en las playas de Ibiza, se contribuye de forma directa a varios de estos ODS, promoviendo una transformación digital orientada al turismo sostenible y a la protección del entorno.

A continuación, se describen los objetivos más directamente relacionados:



Producido en colaboración con TROLLBACK + COMPANY | TheGlobalGoals@trollback.com | +1212.800.1018  
Para cualquier duda sobre la utilización, por favor contactar con: dplcamp@trollback.com

Figura A.1: Resumen de los 17 ODS. Fuente: Página web de la ONU

■ **ODS 9: Industria, innovación e infraestructura.**

Este objetivo busca promover una industrialización inclusiva y sostenible, fomentar la innovación y mejorar la infraestructura tecnológica. La aplicación desarrollada encaja dentro de esta línea al tratarse de una solución tecnológica innovadora, que emplea herramientas de *software* moderno, algoritmos de predicción y conectividad entre servicios para resolver un problema real: la gestión de espacios turísticos en entornos naturales sensibles. Gracias al uso de Inteligencia Artificial, APIs externas y geocalización, se introduce un sistema inteligente que no solo informa, sino que sugiere acciones al usuario, transformando la forma en la que se interactúa con el espacio público.

■ **ODS 11: Ciudades y comunidades sostenibles.**

Uno de los principales pilares del proyecto es fomentar un turismo más equilibrado y menos invasivo. La aplicación permite al usuario informarse en tiempo real del grado de ocupación de las playas, evitando desplazamientos innecesarios, reduciendo el tráfico en zonas costeras y ayudando a distribuir la afluencia de personas de forma más equitativa entre las diferentes playas. Este tipo de herra-

mienta se alinea con la necesidad de diseñar ciudades y destinos más inteligentes, accesibles y respetuosos con el entorno y sus habitantes.

- **ODS 13: Acción por el clima.**

El sistema contribuye también de forma indirecta a la lucha contra el cambio climático. Al reducir el número de desplazamientos innecesarios en vehículo privado, se disminuyen las emisiones de gases de efecto invernadero. Además, al evitar la concentración excesiva de personas en determinadas playas, se limita la presión ambiental en áreas concretas, lo cual favorece una gestión más sostenible del litoral y un uso más racional de los recursos naturales, respetando los ecosistemas locales.

- **ODS 14: Vida submarina.**

Las playas, especialmente aquellas con alto valor ecológico, son espacios frágiles cuyo equilibrio puede verse afectado por una presión turística excesiva. La aplicación ayuda a evitar la sobrecarga en determinadas zonas costeras, lo que indirectamente contribuye a preservar los ecosistemas marinos cercanos. Una menor afluencia incontrolada implica menos residuos, menos alteraciones del hábitat y, en general, una mayor conservación de la biodiversidad marina.



---

---

# APÉNDICE B

---

## MANUAL DE USUARIO DE *S'AFORAPP*

El presente anexo recoge unas instrucciones básicas para el buen uso de la aplicación, así como descripciones breves e imágenes de la misma para que el lector se familiarice con ella.

### **B.1. Introducción**

*S'AforApp* es una aplicación web diseñada para consultar en tiempo real el nivel de ocupación de las playas en Ibiza. Permite a los usuarios participar activamente votando el estado de las playas, recibir recomendaciones alternativas y prever la hora estimada de llenado según condiciones climáticas y temporales.

Su objetivo es mejorar la experiencia del visitante en la isla y fomentar un turismo más sostenible.



Figura B.1: Pantalla de inicio dinámica. Fuente: Elaboración propia

## B.2. Acceso a la Aplicación

Esta sección se va a dividir en dos subsecciones, la de registro (subsección B.2.1) que detalla como dar de alta a un usuario nuevo en la aplicación por primera vez y la de inicio de sesión (subsección B.2.2) que muestra cómo iniciar sesión una vez el usuario está registrado.



Figura B.2: Pantalla con opciones de 'Inicio de Sesión' y 'Registro'. Fuente: Elaboración propia

### B.2.1. Registro de Usuario

Para realizar el registro del usuario, se van a tener que completar todos los campos indicados en el formulario de registro, debiendo cumplir la contraseña con los requisitos establecidos en la parte inferior del formulario y debiendo ser el correo electrónico real.

Inicia sesión aquí'. The footer shows 'S'AforApp © 2025'."/>

Figura B.3: Formulario de registro. Fuente: Elaboración propia

### B.2.2. Inicio de sesión

En cuanto al inicio de sesión, se deberán introducir las credenciales correspondientes al usuario registrado anteriormente, es decir, el correo electrónico y la contraseña que corresponden a ese usuario.

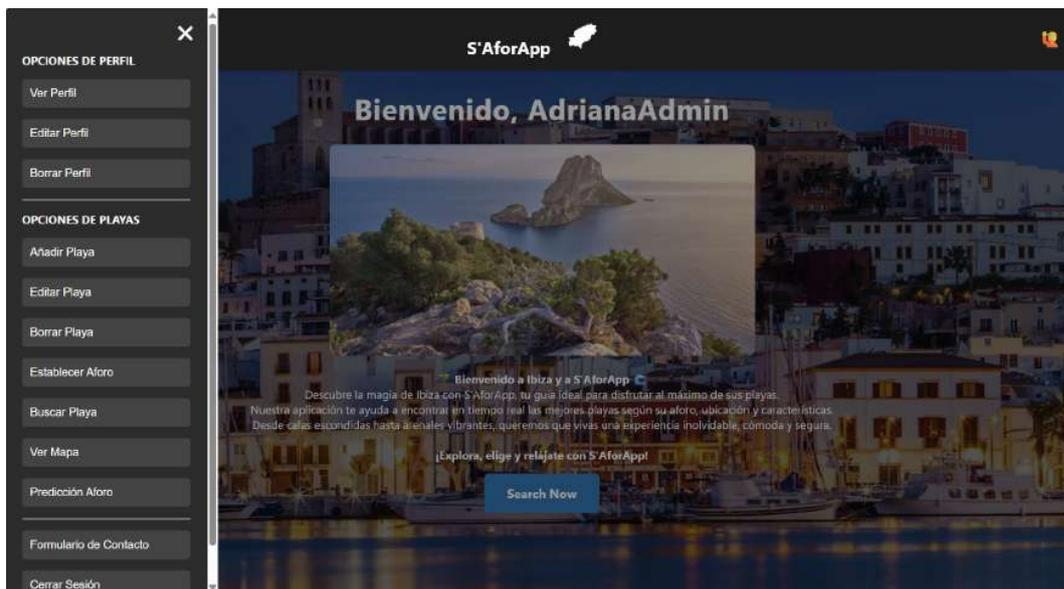
Regístrate aquí'. The footer shows 'S'AforApp © 2025'."/>

Figura B.4: Inicio de sesión. Fuente: Elaboración propia

### B.3. Funcionalidades principales

En este apartado se detallan las funcionalidades principales, mostrando una imagen de como es la pantalla asociada a dicha funcionalidad.

A continuación se muestra la imagen de los dos menús de inicio (usuario y administrador) cuya única variación es el conjunto de funcionalidades disponibles que se encuentran en el menú lateral:



(a) Menú de inicio de Administrador



(b) Menú de inicio de Usuario

Figura B.5: Menús de inicio. Fuente: Elaboración propia

### B.3.1. Consultar Aforo de Playas

Esta es la funcionalidad principal de la aplicación y se trata de la consulta del aforo de las playas. Esto, se muestra de muchas formas y a continuación se va a proceder a la explicación de esta funcionalidad así como de otras que derivan de ésta. Para ello, se pulsa en el botón *'Search Now'* del menú de inicio o en el de *'Buscar Playa'* de la sección lateral, los cuales nos llevarán ambos a la misma página.

#### B.3.1.1. Búsqueda de playas y filtrado de playas

Al entrar en la página, el navegador pedirá acceder a la ubicación actual del dispositivo, lo recomendable es aceptarlo ya que si no se acepta, no se podrán disfrutar todas las funcionalidades de la aplicación relacionadas con la geolocalización.

Una vez aceptado, saldrán todas las playas ordenadas por cercanía al dispositivo desde el cual se está utilizando la ubicación, de esta forma se podrán consultar todas las playas en la base de datos.

También, se podrá filtrar la búsqueda por zona o características de la playa mostrándose así solo las playas correspondientes al filtro en orden de cercanía. Todas estas playas, con o sin filtro aplicado, se mostrarán también en un mapa en la zona inferior de la página.

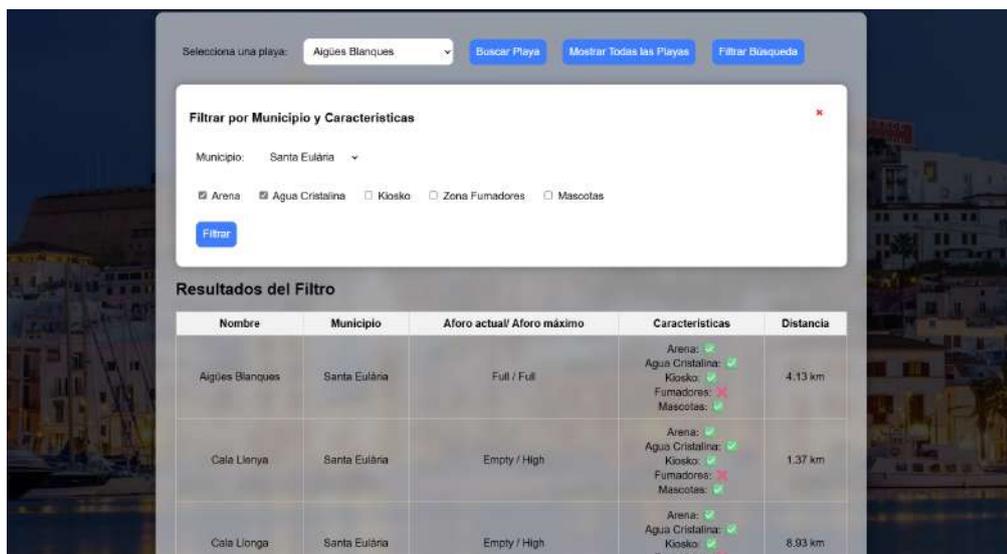


Figura B.6: Filtro del buscador de playas. Fuente: Elaboración propia

Finalmente, existe un desplegable con todas las playas de la base de datos, que dará la opción de seleccionar una de ellas y, pulsando el botón ‘Buscar Playa’, el sistema proporcionará la información de aquella playa buscada y la indicará en el mapa. En el caso de que esa playa se encuentre en su aforo máximo se realizará una recomendación de aquellas playas con más similitudes a las buscadas.

En caso de querer visualizar las playas por cercanía como al principio, es suficiente con pulsar el botón ‘Mostrar Todas las Playas’.

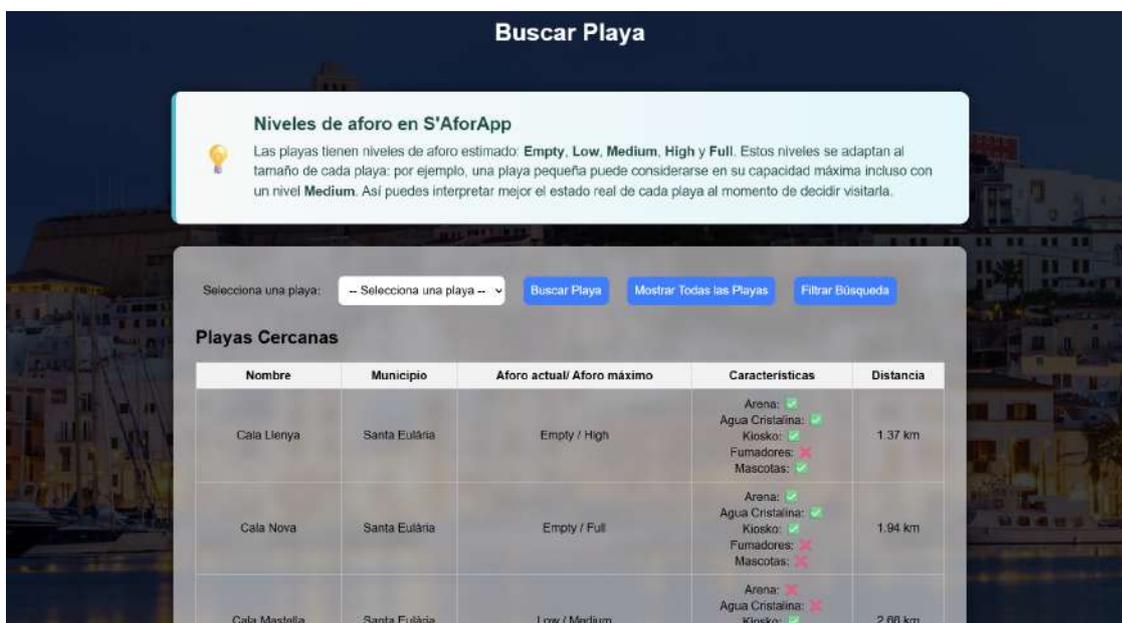
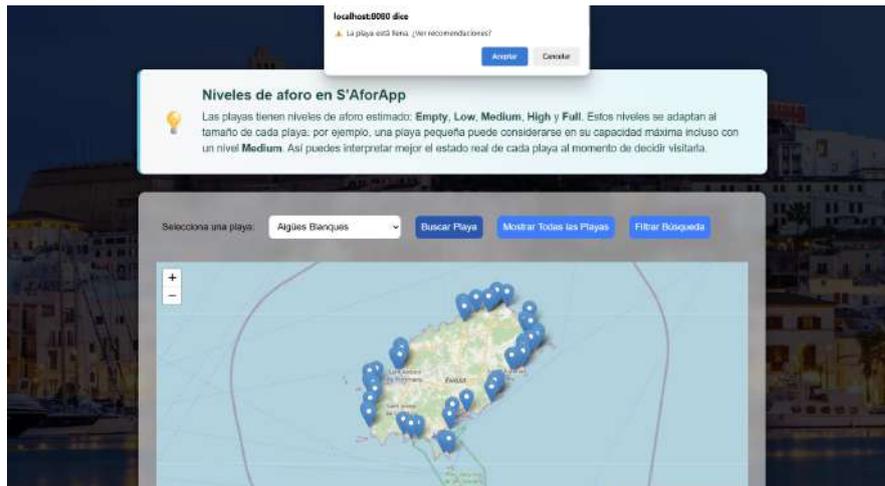


Figura B.7: Buscador de playas. Fuente: Elaboración propia

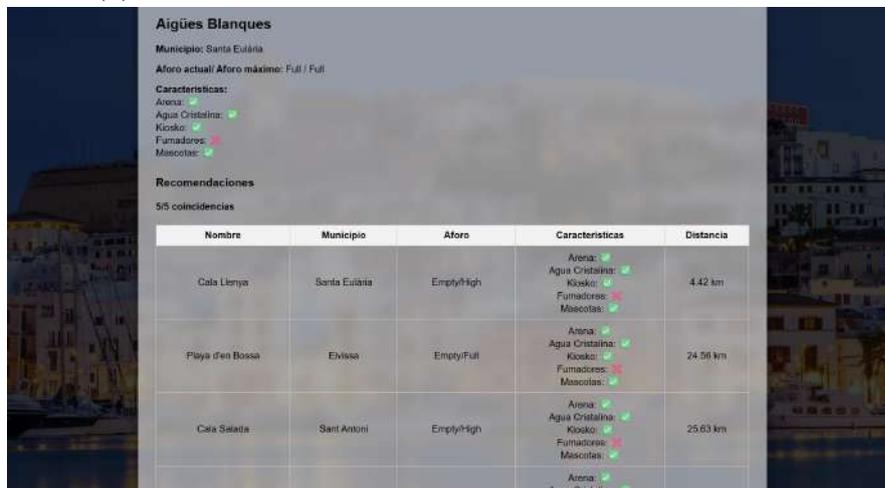
### B.3.1.2. Recomendador de playas (*'Where to go'*)

Cuando la playa consultada se encuentra completa, el sistema activa automáticamente un recomendador que sugiere playas alternativas. Este recomendador compara las características de la playa llena con el resto de playas existentes, y muestra una lista ordenada según el grado de similitud, priorizando aquellas con más coincidencias.

Además, dentro de cada grupo de coincidencias, las playas se ordenan por cercanía respecto a la playa original. La tabla resultante incluye toda la información detallada sobre cada playa.



(a) Mensaje de 'playa llena'. Fuente: Elaboración propia



(b) Recomendador de playas. Fuente: Elaboración propia

Figura B.8: Recomendador de playas. Fuente: Elaboración propia

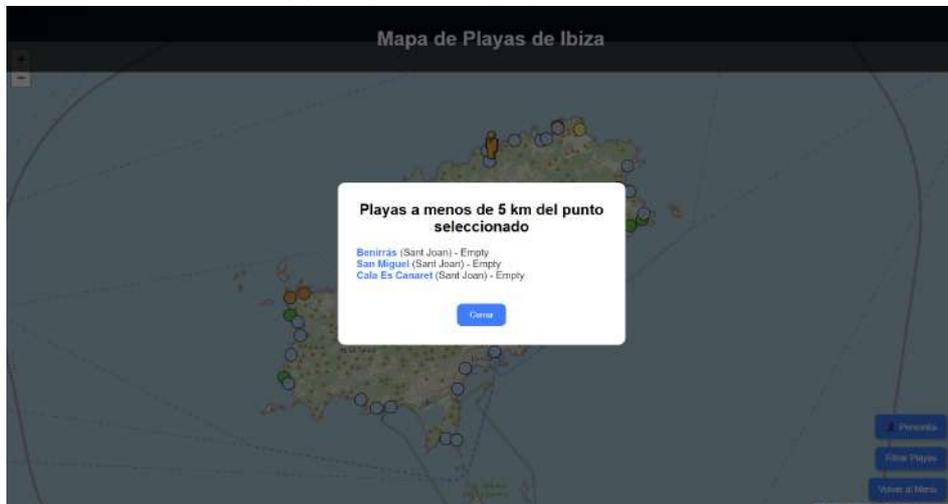
### B.3.1.3. Visualización de mapa

Otra forma de visualizar las playas es mediante un mapa interactivo que tiene distintas opciones entre las cuales se encuentra el filtrado, que sigue el mismo procedimiento que el filtrado anterior y, la funcionalidad de mover una 'personita' sobre el mapa y que muestre las playas que se encuentran a 5km del punto seleccionado. De esta forma se mostrará una lista de las playas y si se pulsa encima del nombre de alguna de las playas aparecerá un modal con la información de la playa seleccionada.

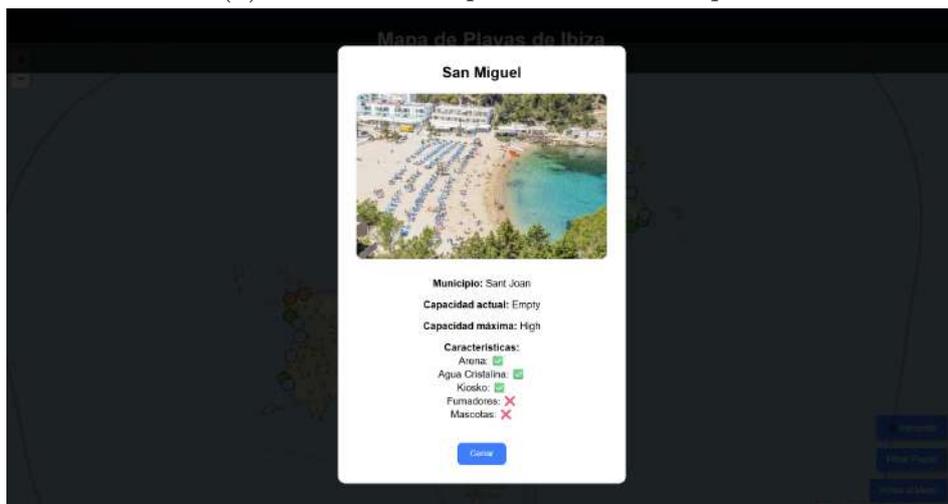
En caso de no querer hacer uso de ninguna de las dos funcionalidades previas, las playas se representarán sobre el mapa con puntos de colores según su aforo y, si se desea visualizar la playa de la que se trata, basta con seleccionar dicho punto.



(a) Mapa de las playas de Ibiza



(b) Funcionalidad 'personita' en el mapa



(c) Panel con la información de la playa seleccionada

Figura B.9: Visualización del mapa de las playas de Ibiza. Fuente: Elaboración propia

### B.3.2. Establecer Aforo

Para establecer un aforo y colaborar en conseguir un mejor funcionamiento de la aplicación, se procederá a votar un aforo para una playa. Para hacerlo, el procedimiento es muy sencillo, aparecerá un desplegable con todas las playas almacenadas en la base de datos y se indicará junto a ellas cual es su capacidad máxima. A continuación, se encontrará otro desplegable que mostrará solo las opciones de aforo posibles, sin mostrar la posibilidad de votar un aforo mayor a su máximo.

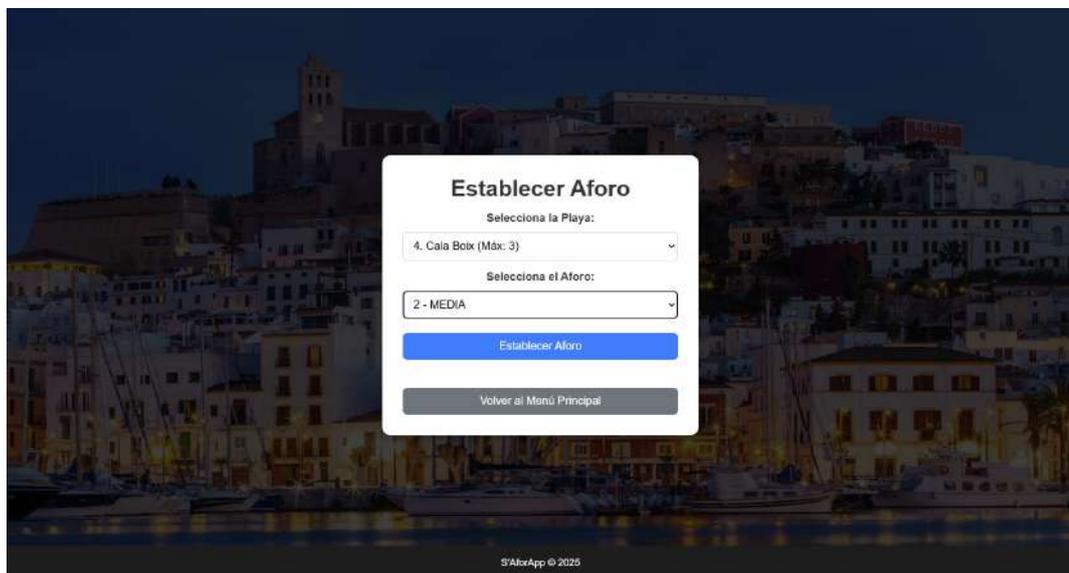
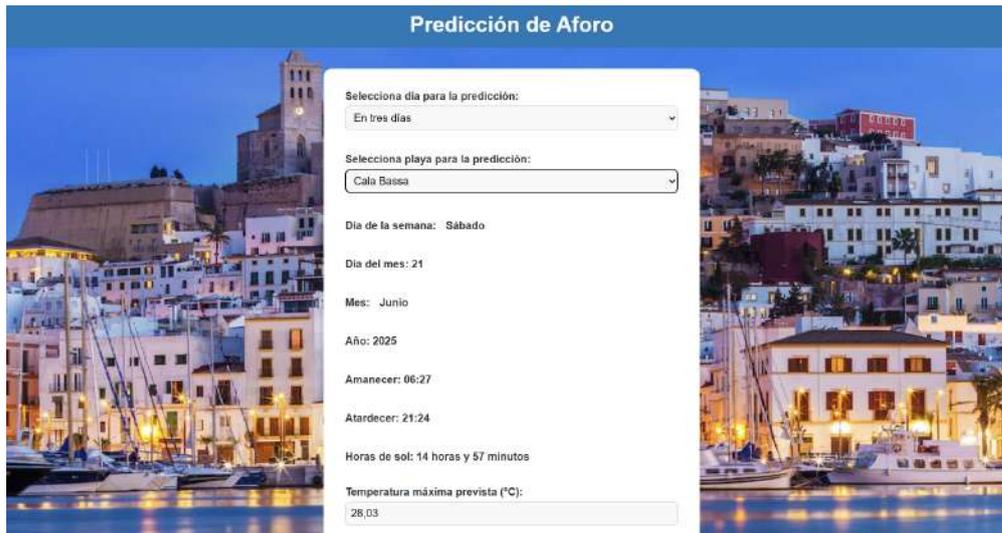


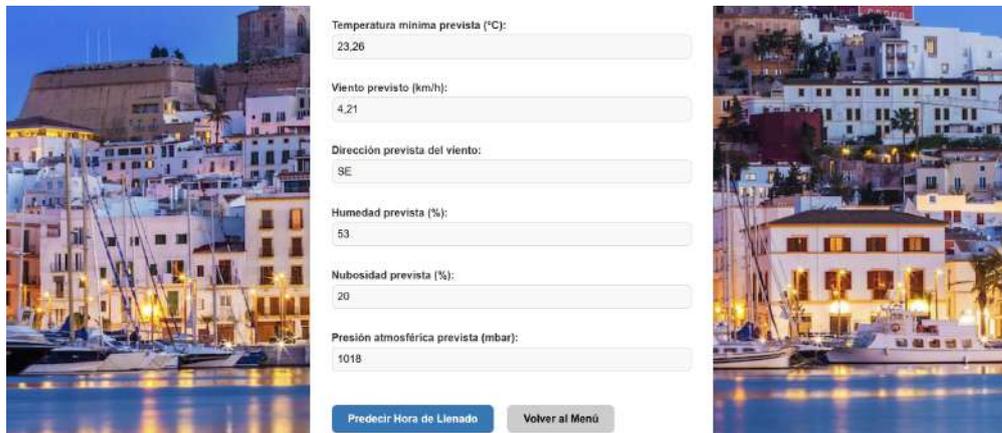
Figura B.10: Funcionalidad de establecer aforo. Fuente: Elaboración propia

### B.3.3. Predicción de hora de llenado

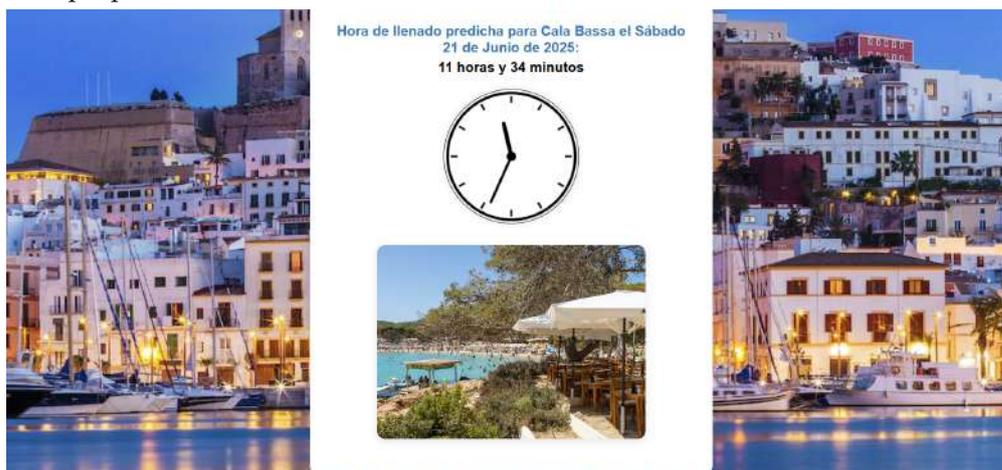
El sistema de predicción de la hora de llenado de las playas funciona de la siguiente forma. A través de un menú desplegable, el usuario puede seleccionar una playa concreta y elegir el día para el que desea obtener la predicción: mañana, pasado mañana o en tres días. Con esta información, el sistema recopila automáticamente datos meteorológicos actualizados, como temperatura, nubosidad, viento, y también factores astronómicos como la hora de salida y puesta del sol, obtenidos de APIs externas. Todos estos datos, junto con registros históricos, se procesan mediante un modelo de *Machine Learning* previamente entrenado, que devuelve como resultado la hora estimada, mostrada en un reloj, en la que esa playa alcanzará su máxima ocupación (figura B.11c).



(a) Página de predicción de hora de llenado de las playas (I). Fuente: Elaboración propia



(b) Página de predicción de hora de llenado de las playas (II). Fuente: Elaboración propia



(c) Página de predicción de hora de llenado de las playas (III). Fuente: Elaboración propia

Figura B.11: Predicción de la hora de llenado de las playas mediante un modelo de *Machine Learning*. Fuente: Elaboración propia

### **B.3.4. Gestión del perfil**

Dentro de la gestión del perfil podemos diferenciar 3 funcionalidades muy simples y fáciles de llevar a cabo: ‘visualizar perfil’ (subsección B.3.4.1), ‘editar perfil’ (subsección B.3.4.2) y ‘borrar perfil’ (subsección B.3.4.3).

#### **B.3.4.1. Visualizar perfil**

Para la visualización del perfil solo hace falta pulsar sobre el botón del menú lateral que pone ‘Ver Perfil’. A continuación se adjunta una imagen de la página que aparecerá al pulsarse dicho botón.

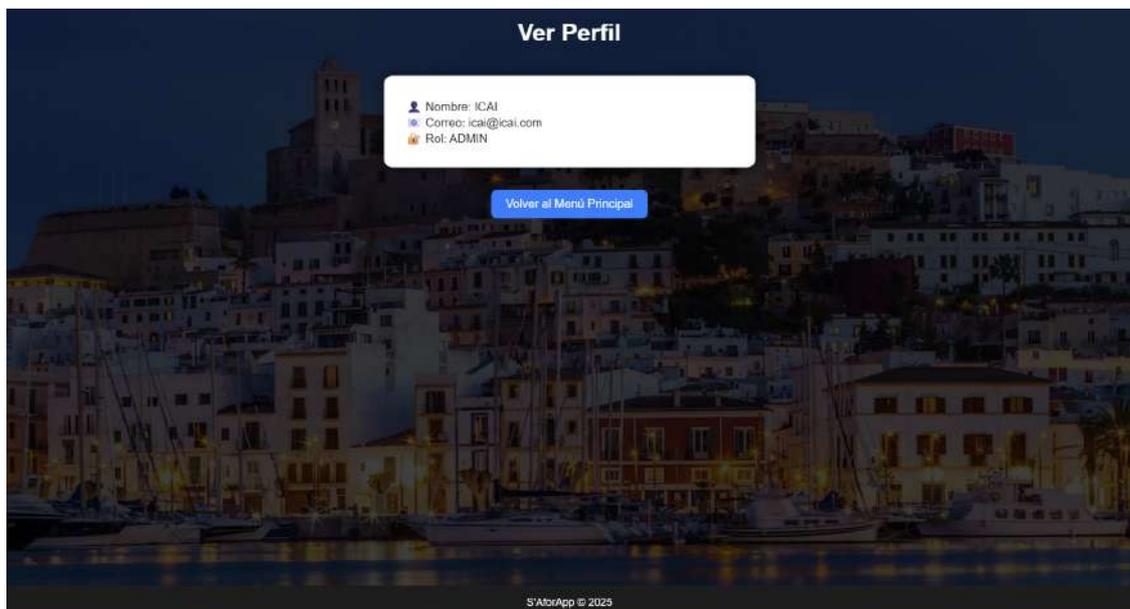


Figura B.12: Página de visualización del perfil. Fuente: Elaboración propia

#### **B.3.4.2. Editar perfil**

Para la edición del perfil solo hace falta pulsar sobre el botón del menú lateral que pone ‘Editar Perfil’. Una vez se abra la página correspondiente, se podrán editar los campos que se permitan y posteriormente darle al botón inferior para que así, la nueva información se almacene correctamente en la base de datos. A continuación, se adjunta una imagen de la página que contiene el formulario de edición del perfil.

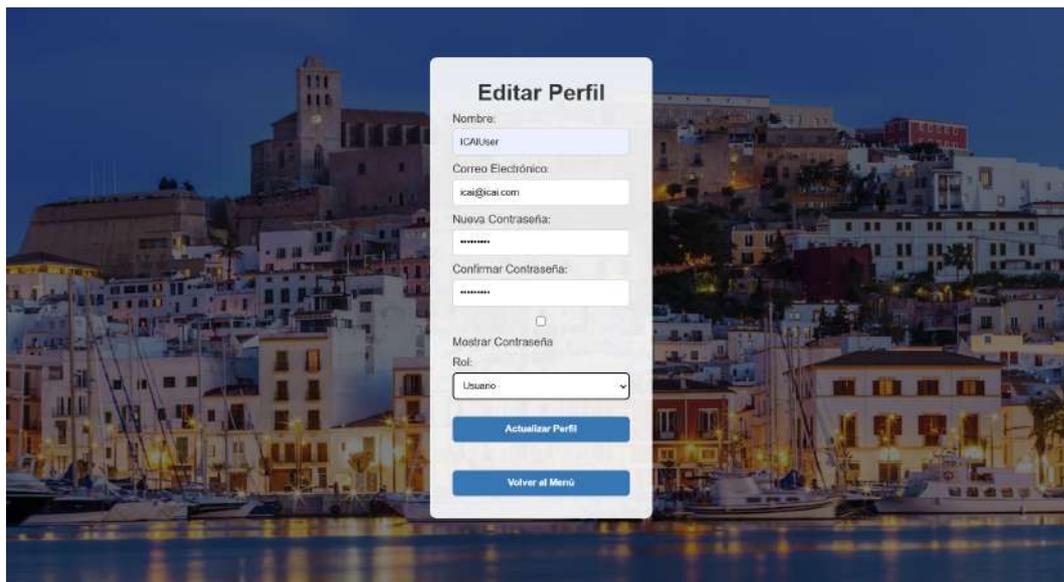


Figura B.13: Página de edición del perfil. Fuente: Elaboración propia

#### B.3.4.3. Borrar perfil

Para el borrado del perfil solo hace falta pulsar sobre el botón del menú lateral que pone 'Borrar Perfil'. Una vez se pulse, se volverá a preguntar si se desea realizar el borrado y en caso de confirmar, la cuenta se borrará correctamente, redirigiéndose a la página de inicio de sesión. A continuación, se adjunta una imagen de la página en la que se realiza el borrado del perfil.

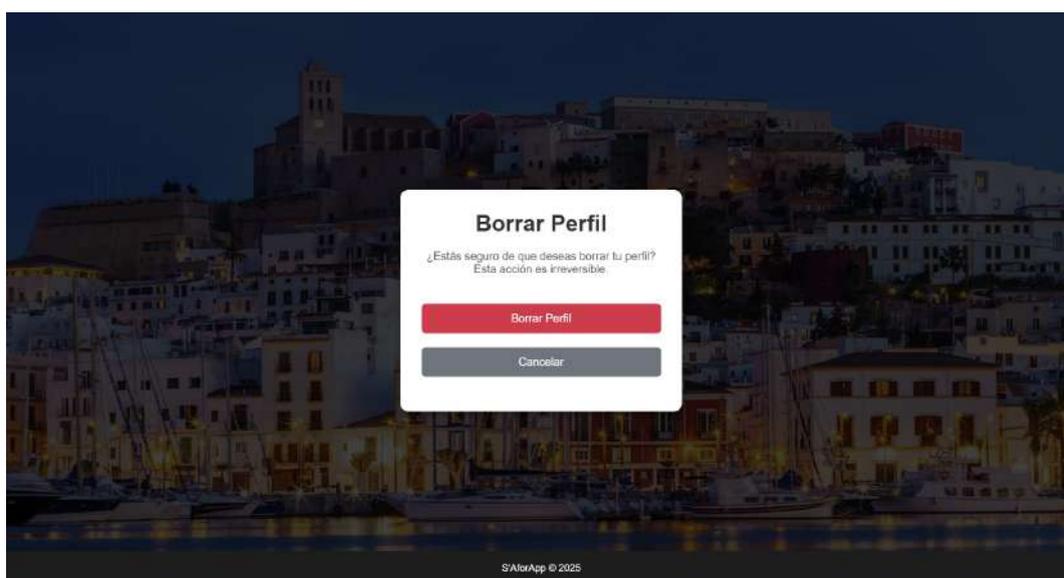


Figura B.14: Página de borrado del perfil. Fuente: Elaboración propia

### B.3.5. Formulario de contacto

Para que el usuario pueda dar su opinión o ponerse en contacto con el equipo de *S'AforApp*, se ha creado un formulario de contacto al que se puede acceder pulsando el botón del menú lateral 'Formulario de Contacto'. A continuación, se dirigirá a una página con un formulario a rellenar y posteriormente enviar pulsando el botón de la parte inferior de la página. En caso de haberse enviado correctamente, se mostrará un mensaje de confirmación y recibirá en su *email* una copia de aquello que ha rellenado.

(a) Página del formulario de contacto (I). Fuente: Elaboración propia

(b) Página del formulario de contacto (II). Fuente: Elaboración propia

(c) Página del formulario de contacto con mensaje de **OK**. Fuente: Elaboración propia

Figura B.15: Formulario de contacto para recibir *feedback* del usuario. Fuente: Elaboración propia

### **B.3.6. Cerrar sesión**

Para cerrar sesión, simplemente presiona el botón del menú lateral que pone ‘Cerrar Sesión’. El usuario será redirigido a la página de inicio de sesión con la correspondiente eliminación del *token* de sesión.

## **B.4. Funciones avanzadas (únicamente para administradores)**

Hay ciertas funcionalidades que solo pueden ser realizadas por administradores, para evitar que usuarios estándares realicen operaciones que puedan comprometer las funcionalidades de la aplicación. En esta sección se detallan cada una de ellas y están relacionadas con la adición (subsección B.4.1.1), la modificación (subsección B.4.1.2) de la información de las playas o la eliminación (subsección B.4.1.3) de playas del sistema.

### **B.4.1. Gestión de playas**

La gran diferencia en cuanto al usuario y al administrador se encuentra en que el usuario no puede llevar a cabo ninguna funcionalidad de gestión de playas. Estas están ideadas exclusivamente para el uso de los administradores.

#### **B.4.1.1. Añadir playa**

Para añadir una playa, se debe pulsar en el menú lateral el botón ‘Añadir Playa’ y, posteriormente, rellenar toda la información de la playa que se desea añadir. Una vez completado, se enviará el formulario pulsando el botón inferior y la nueva playa se almacenará en la base de datos.

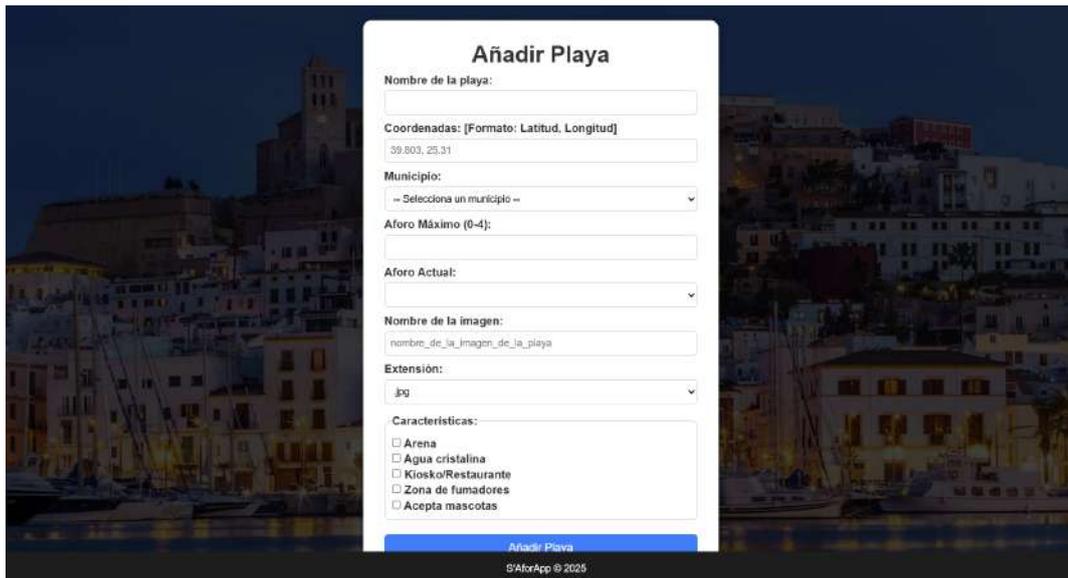


Figura B.16: Página de adición de playas. Fuente: Elaboración propia

#### B.4.1.2. Modificar playa

Para modificar los datos de una playa se deberá pulsar el botón del menu lateral 'Editar Playa', este llevará a un formulario similar al de 'Añadir Playa' pero en este caso habrá campos en los que no se podrá escribir, es decir, no serán editables.

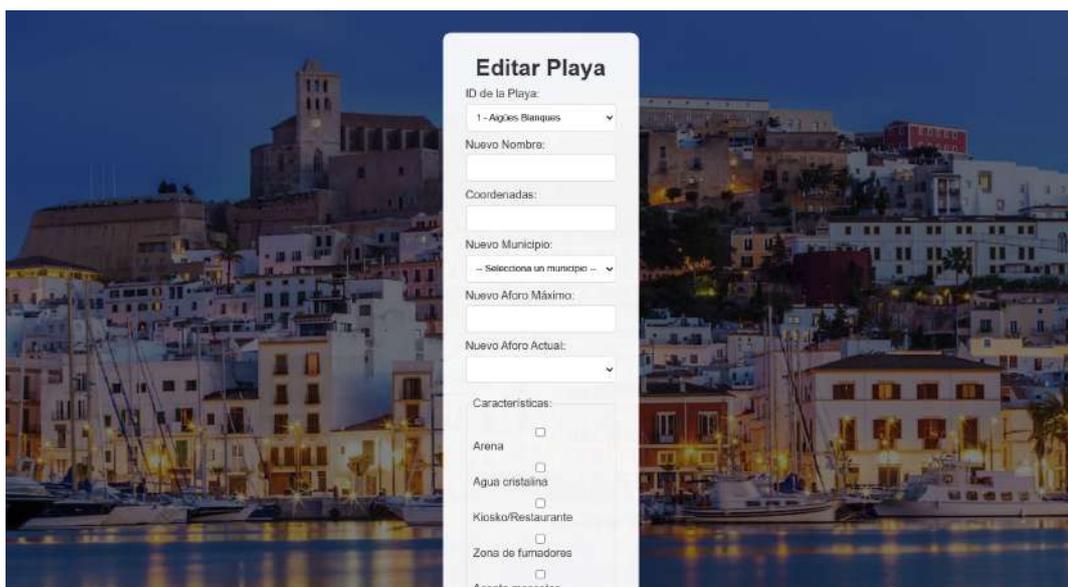


Figura B.17: Página de edición de playas. Fuente: Elaboración propia

### B.4.1.3. Eliminar playa

La funcionalidad de eliminar playa es muy sencilla de llevar a cabo, al igual que en las funcionalidades anteriores, el botón para acceder a la página correspondiente se encuentra en el menú lateral y se llama 'Borrar Playa'. Una vez se pulse se abrirá la página que contendrá un desplegable con todas las playas. Se seleccionará la playa que se desea borrar y se pulsará el botón de borrado. De esta forma, la playa se borrará también de la base de datos.

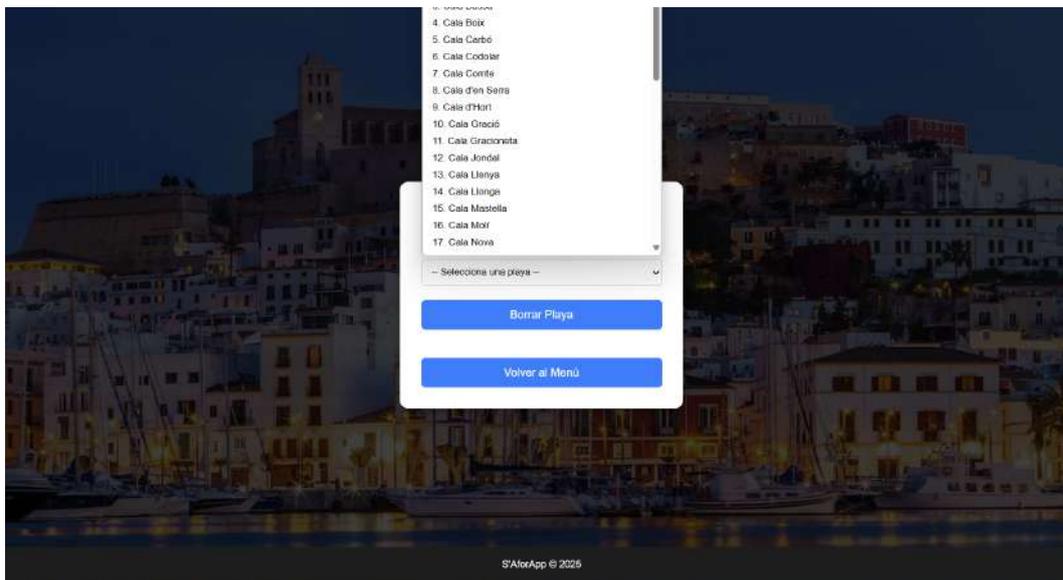


Figura B.18: Página de borrado de playas. Fuente: Elaboración propia

### B.4.2. Editar perfil (tipo de usuario)

En cuanto a la edición de perfil del administrador, hay una ligera diferencia con el del usuario y es que el administrador podrá cambiar su rol en la aplicación al de usuario estándar pero el usuario no podrá elegir ser administrador.

## B.5. Recomendaciones de uso

Para contribuir al buen funcionamiento del sistema, se recomienda que los usuarios participen activamente votando el nivel de ocupación únicamente cuando estén físicamente en la playa, lo que ayuda a mejorar la precisión y fiabilidad de los datos.

Además, se aconseja utilizar el sistema de recomendaciones para descubrir playas menos

concurridas, especialmente en momentos de alta ocupación.

Por último, si se planea una visita con antelación, es útil consultar la funcionalidad de predicción de aforo, que permite anticiparse a las horas de mayor afluencia y organizar mejor la jornada.

## **B.6. Soporte**

Para cualquier consulta o incidencia, se puede escribir a: **[sforappcompany@gmail.com](mailto:sforappcompany@gmail.com)** o utilizar el formulario de contacto incluido en la aplicación.



---

# BIBLIOGRAFÍA

- Ajuntament de Palma (2020). L'ajuntament llança una nova app emmarcada en el projecte platges segures. Accedido: 2025-06-09.
- Apache Software Foundation (2025). Apache maven project.
- Ayuntamiento de Málaga (2023). Playasapp: Consulta el estado de las playas.
- Cinco Días (2025). Ibiza imita a formentera y limita la entrada de coches para combatir el turismo masivo.
- Cricelli, L., Grimaldi, M., and Vermicelli, S. (2022). Crowdsourcing and open innovation: a systematic literature review, an integrated framework and a research agenda. *Review of Managerial Science*, 16(5):1269–1310.
- Eclipse Foundation (2024). Jakarta bean validation.
- Fernández-Pacheco Sánchez-Migallón, A. R. (2022-2023). Apuntes de la asignatura de 'ingeniería de software'.
- Fernández-Pacheco Sánchez-Migallón, A. R. (2023-2024). Apuntes de la asignatura de 'programación de aplicaciones telemáticas'.
- Ferrer, M. A. (2023). La experiencia negativa de la masificación turística. el caso de las islas baleares. *Estudios Políticos y Sociales Internacionales*, (5):45–60.
- Govern Balear (2020). Platges segures: Aplicació de control d'aforament de platges.
- Granados, B. (2024). Análisis del turismo masivo en ibiza en 2024 y el impacto en sus residentes.
- Institut de Ciència i Tecnologia Ambientals (ICTA-UAB) (2021). El turismo es el principal responsable de la basura marina en las playas del mediterráneo.
- Instituto Nacional de Estadística (2024). Viajeros y pernoctaciones por zonas turísticas. Recuperado de Internet.
- Kanhere, S. S. (2013). Participatory sensing: Crowdsourcing data from mobile smartphones in urban spaces. *IEEE Internet Computing*, 17(4):30–36.

## BIBLIOGRAFÍA

---

- Luo, Y., Tang, J., and Wu, Y. (2020). Pedestrian flow prediction with recurrent neural networks in public spaces. *Sensors*, 20(3):687.
- Naciones Unidas (2015). La asamblea general adopta la agenda 2030 para el desarrollo sostenible.
- Periódico de Ibiza (2022). El consell instala cámaras de seguridad para controlar el aforo en playas de ibiza.
- Pivotal Software (2024). Spring boot documentation.
- Scikit-learn Developers (2024). Model persistence.
- Spring Team (2025a). Spring boot - spring framework.
- Spring Team (2025b). Spring security - spring framework.
- The Apache Software Foundation (2024). Maven project.
- Thomas Mueller (2025). H2 database engine.
- Vodafone España (2022). Ibiza instala cámaras con ia para controlar el aforo de 33 playas.